Distribution Agreement

Signature:

________________________________ ____________

Yuyang Gao Date

Interpretable and Interactive Representation Learning on Geometric Data

By

Yuyang Gao
Doctor of Philosophy

Computer Science and Informatics

Liang Zhao, Ph.D.
Advisor

Carl Yang, Ph.D.
Committee Member

Joyce C. Ho, Ph.D.
Committee Member

Lingfei Wu, Ph.D.
Committee Member

Accepted:

Kimberly R.J. Arriola, Ph.D.
Dean of the James T. Laney School of Graduate Studies

Date

Interpretable and Interactive Representation Learning on Geometric Data

By

Yuyang Gao
B.S., Shandong University, Shandong, China, 2014
M.S., George Mason University, VA, 2018

Advisor: Liang Zhao, Ph.D.

An abstract of
A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Computer Science and Informatics
2022

Abstract

Interpretable and Interactive Representation Learning on Geometric Data
By Yuyang Gao

In recent years, representation learning on geometric data, such as image and graph-structured data, are experiencing rapid developments and achieving significant progress thanks to the rapid development of Deep Neural Networks (DNNs), including Convolutional Neural Networks (CNNs) and Graph Neural Networks (GNNs). However, DNNs typically offer very limited transparency, imposing significant challenges in observing and understanding when and why the models make successful/unsuccessful predictions [61]. While we are witnessing the fast growth of research in local explanation techniques in recent years, the majority of the focus is rather handling "how to generate the explanations", rather than understanding "whether the explanations are accurate/reasonable", "what if the explanations are inaccurate/unreasonable", and "how to adjust the model to generate more accurate/reasonable explanations" [13, 108, 168, 88, 129, 130, 62, 183, 151].

To explore and answer the above questions, this dissertation aims to explore a new line of research called 'Explanation-Guided Learning' (EGL) that intervenes the deep learning models' behavior through XAI techniques to jointly improve DNNs in terms of both their explainability and generalizability. Particularly, we propose to explore the EGL on geometric data, including image and graph-structured data, which are currently under-explored [61] in the research community due to the complexity and inherent challenges in geometric data explanation.

To achieve the above goals, we start by exploring the interpretability methods for geometric data on understanding the concepts learned by the deep neural networks (DNNs) with bio-inspired approaches and propose methods to explain the predictions of Graph Neural Networks (GNNs) on healthcare applications. Next, we design an interactive and general explanation supervision framework GNES for graph neural networks to enable the "learning to explain" pipeline, such that more reasonable and steerable explanations could be provided. Finally, we propose two generic frameworks, namely GRADIA and RES, for robust visual explanation-guided learning by developing novel explanation model objectives that can handle the noisy human annotation labels as the supervision signal with a theoretical justification of the benefit to model generalizability.

This research spans multiple disciplines and promises to make general contributions in various domains such as deep learning, explainable AI, healthcare, computational neuroscience, and human-computer interaction by putting forth novel frameworks that can be applied to various real-world problems where both interpretability and task performance are crucial.

Interpretable and Interactive Representation Learning on Geometric Data

By

Yuyang Gao
B.S., Shandong University, Shandong, China, 2014
M.S., George Mason University, VA, 2018

Advisor: Liang Zhao, Ph.D.

A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Computer Science and Informatics
2022

Acknowledgments

First, I would like to express my sincere gratitude to to my advisor, Dr. Liang Zhao, for his support during my Ph.D. study. I have been incredibly fortunate to benefit from his insightful and patient guidance on doing research. His immense knowledge and plentiful experience have encouraged me in all the time of my academic research and daily life, especially the hardest moments of my PhD years. I will never forget his great efforts on refining and commenting my research papers and slides again and again. Without his guidance, I would not have made it.

I am very thankful to all my committee members, who have provided not only important guidance for the completion of my dissertation, but also for my research work during our collaborations. Thanks go to Dr. Lingfei Wu who provided extensive expertise in Natural Language Processing that greatly support my health care related research projects. I am impressed and learnt a lot from her high efficiency and deep insights in scientific research. Thanks also go to Dr. Carl Yang and Dr. Joyce Ho for providing insightful comments and valuable suggestions from my preliminary proposal, through my research defense, and for my final defense. Special thanks go to Dr. Sungsoo Ray Hong who provided extensive expertise in Human Computer Interaction that opened up the new research windows on Explanation-Guided Learning (EGL) that requires interdisciplinary knowledge and collaboration.

I would like to also express appreciation to my friends in and outside my Lab for our unforgettable collaborations: Tong Steven Sun, Dazhou Yu, Xiaojie Guo, Guangji Bai, Siyi Carrie Gu, Tanmoy Chowdhury, Negar Etemadyrad, Qingzhe Li, Junxiang Wang, to name but a few-with whom I have shared unique moments throughout this time. Finally, I would like to dedicate this dissertation with special thanks to my parents, my dear brothers and sisters from Church in Atlanta and Church in Dunn Loring. These great people have provided me with enormous love, courage, support throughout this process.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

As Deep Neural Networks (DNNs) are widely deployed in sensitive application areas, recent years have seen an explosion of research in understanding how DNNs work under the hood (e.g., explainable AI, or XAI) [8, 5] and more importantly, how to improve DNNs using human knowledge [61]. In particular, representation learning on geometric data, such as image and graph-structured data have been increasingly grabbed attention in several research fields, including computer vision [108, 43], natural language processing [7], medical domain [33], and beyond. Such trend is attributed to the practical implication of geometric data—many real-world data, such as social networks [40], chemical molecules [128], and financial data [96], are represented as image or graphs.

In recent years, representation learning on geometric data, such as image and graph-structured data, are experiencing rapid developments and achieving significant progress thanks to the rapid development of Deep Neural Networks (DNNs), including Convolutional Neural Networks (CNNs) and Graph Neural Networks (GNNs). However, DNNs typically offer very limited transparency, imposing significant challenges in observing and understanding when and why the models make successful/unsuccessful predictions [61, 159]. This issue motivates a surge of recent research on Explainable Artificial Intelligence (XAI) techniques, including gradients-based methods, where the gradients are used to indicate the importance of different input features [13, 108]; perturbation-based methods, where an additional optimization step is typically used to find the important input that influences the model output the most with input perturbations [168, 88, 129]; response-based methods, where the output response signal is backpropagated as an importance score layer by layer until the input space [13, 108, 130]; surrogate-based methods, where the explanation obtained from an interpretable surrogate model that is trained to fit the original prediction is used to explain the original model [62, 183, 151]; and global explanation methods, where graph patterns are generated to maximize the predicted probability for a certain class and use such graph patterns to explain the class [173].

Despite the recent fast progress on explanation techniques for DNNs, the majority of the research body in XAI put focus on handling “how to generate the explanations” while showing less attention to advanced questions like “whether the explanations are reasonable/accurate”, “what if the explanations are unreasonable/inaccurate”, and most importantly, “how to adjust the model to generate more reasonable/accurate explanations in the future”. We argue that understanding how to convert insights learned from XAI-driven techniques to steer DNNs would be the key to realizing the DNNs to be more powerful, fair, accountable, transparent, unbiased, and trustworthy, unraveling many real-world application areas.

Recently, a new line of research named Explanation-Guided Learning (EGL) [82, 125, 63, 140] that aims to intervene ML model’s behavior through XAI techniques has started to emerge. In particular, the approaches jointly improve DNNs in terms of both their explainability and generalizability by applying additional supervision signals or prior knowledge onto the model reasoning process to direct the model explanation derived from established XAI techniques. Despite the fact that EGL techniques are generally still in their early stage, the majority of existing studies have produced encouraging results, showing that the main DNNs can generally benefit from the additional explanation objective in terms of both model explainability and generalizability to unseen data across various application domains. However, developing EGL frameworks can be difficult for geometric data due to several challenges: **1) Difficulty in enhancing the explainability of geometric neural network models.** Due to the complex data structure and high dimensionality of geometric data, it is difficult to directly apply conventional explainability techniques to geometric neural network models, such as CNNs and GNNs. Thus enhancing the explainability of geometric neural networks models can be particularly helpful for representation learning on geometric data, especially for graph-structured data, as non-expert humans cannot intuitively understand the relevant context within a network, for example, when identifying groups of atoms (a sub-graph structure on a molecular graph) that contribute to a particular property of a molecule [108]. **2) Difficulty in refining the geometric neural networks model's explanation.** For graph-structured data, existing GNN explanation works usually focus on either node and edge explanation while the interplay and consistency between the explanations of nodes and edges are extremely challenging to maintain and jointly adjusted; while for image data, some important object parts or even the entire objects may be missed by the coarsely drawn boundary from human annotators. Thus, applying naive supervision directly to train the model can lead to falsely excluding non-trivial features from the input space that are important to the prediction. **3) Difficulty in jointly improving model performance and explainability with limited explanation supervision.** Due to the high cost of human annotation, it can be impractical to assume full accessibility to the human explanation label during model training. Thus designing an effective framework that can best leverage a partially labeled dataset is on-demand yet challenging.

The potential of applying supervision to improve the model's explanation in DNNs

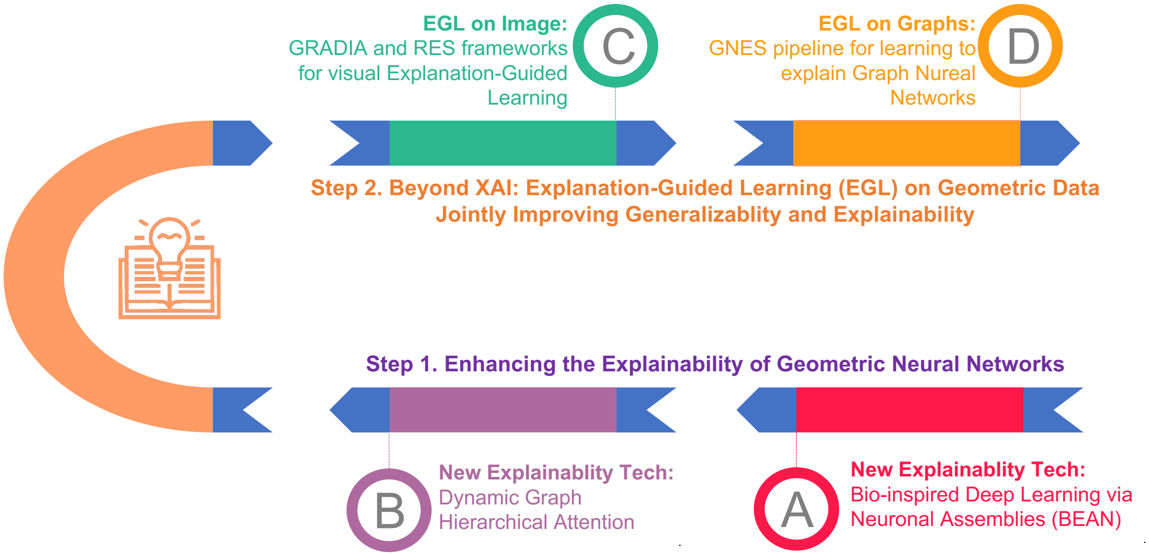

EGL on Image:
GRADIA and RES frameworks for visual Explanation-Guided Learning
C
EGL on Graphs:
GNES pipeline for learning to explain Graph Nureal Networks
D
Step 2. Beyond XAI: Explanation-Guided Learning (EGL) on Geometric Data
Jointly Improving Generalizablity and Explainability
Step 1. Enhancing the Explainability of Geometric Neural Networks
B
New Explainablity Tech:
Dynamic Graph Hierarchical Attention
A
New Explainablity Tech:
Bio-inspired Deep Learning via Neuronal Assemblies (BEAN)

Figure 1.1: The Overview Flowchart of My Research Studies

has been studied in many domains across different applications, such as texts [65, 119], and attributed data [150], and more. However, the research on supervising explanations on image data—where the explanation is represented through saliency maps—is still under-explored [61]. In part, this is due to several inherent challenges in supervising visual explanations. Moreover, EGL on graph-structured data with graph neural networks has not yet been explored before.

Therefore, the goal of my research explores two important problems for geometric data, namely **1) how to enhance the explainability of geometric neural networks, including CNNs and GNNs**, and **2) how explainability can further benefit model's generalizability**, as shown in details in Figure 1.1. The details of each research issue are provided in the following subsections.

1.1 Research Issues

This research aims on the exploration of designing an interpretable and interactive learning framework for geometric neural network models as well as the applications to the real-world tasks. As illustrated in Figure 1.1, the major research issues can be stated as follows:

1.1.1 Interpretable and Efficient Bio-inspired Deep Learning via Neuronal Assemblies

Deep neural networks (DNNs) are known for extracting useful information from large amounts of data. However, the representations learned in DNNs are typically hard to interpret, especially for high dimensional geometric data, such as images and graphs. One crucial issue of the classical DNN model such as multilayer perceptron (MLP) is that neurons in the same layer of DNNs are conditionally independent of each other, which makes co-training and emergence of higher modularity difficult. In contrast to DNNs, biological neurons in mammalian brains display substantial dependency patterns. Specifically, biological neural networks encode representations by so-called neuronal assemblies: groups of neurons interconnected by strong synaptic interactions and sharing joint semantic content. The resulting population coding is essential for human cognitive and mnemonic processes. Here, we propose a novel Biologically Enhanced Artificial Neuronal assembly (BEAN) [45] regularization to model neuronal correlations and dependencies, inspired by cell assembly theory from neuroscience. Experimental results show that BEAN enables the formation of interpretable neuronal functional clusters and consequently promotes a sparse, memory/computation-efficient network without loss of model performance. Moreover, our few-shot learning experiments demonstrate that BEAN could also enhance the generalizability of the model when training samples are extremely limited.

1.1.2 Interpretation for Dynamic Attributed Graphs via Hierarchical Attention

Online health communities such as the online breast cancer forum enable patients (i.e., users) to interact and help each other within various subforums, which are subsections of the main forum devoted to specific health topics. The changing nature of the users' activities in different subforums can be strong indicators of their health stages or changes in their treatment changes. This additional patient information could allow health-care organizations to respond promptly and provide valuable additional information for each patient's specific health stage. However, modeling complex dynamic transitions of an individual user's activities among different subforums over time and learning how these correspond to his/her health stage are extremely challenging problems that cannot be addressed by existing methods. In this thesis, we first formulate the transition of user activities as a dynamic graph with multi-attributed nodes, then formalize the health stage inference task as a dynamic graph-to-sequence learning problem, and hence propose novel and generic dynamic graph-to-sequence neural networks architecture (DynGraph2Seq) to address all the challenges [44]. Our proposed DynGraph2Seq model consists of a novel dynamic graph encoder and an interpretable sequence decoder that learn the mapping between a sequence of time-evolving user activity graphs and a sequence of target health stages. We go on to propose new dynamic graph regularization and dynamic graph hierarchical attention mechanisms to facilitate the necessary multi-level interpretability. A comprehensive experimental analysis of its use for a health stage prediction task demonstrates both the effectiveness and the interpretability of the proposed models.

1.1.3 Explanation-Guided Representation Learning on Geometric Data

In recent years, convolutional neural networks (CNNs) and graph neural networks (GNNs) and the research on their explainability are experiencing rapid developments and achieving significant progress. Many methods are proposed to explain the predictions of CNNs and GNNs, focusing on "how to generate explanations". However, research questions like "whether the GNN explanations are inaccurate", "what if the explanations are inaccurate", and "how to adjust the model to generate more accurate explanations" have not been well explored. To address the above questions, we aim to propose generic pipelines and frameworks to adaptively learn how to explain GNNs and CNNs more accurately and effectively on graph-structured and image data, respectively. Specifically, for handling GNNs on graph-structured data, we propose a novel GNN Explanation Supervision (GNES) [46] framework that can jointly optimize both model prediction and explanation by enforcing both whole graph regularization and weak supervision on model explanations. For the graph regularization, our intention is to propose a unified explanation formulation for both node-level and edge-level explanations by enforcing the consistency between them. For CNNs on image data, we propose two EGL frameworks, namely GRADIA [48] and RES [47], that enable explanation supervision on DNNs that can handle both positive and negative explanation annotation labels with a novel robust explanation loss that is designed to handle the inaccurate boundary, incomplete region, as well as inconsistent distribution challenges in applying the noisy human annotation labels as the supervision signal. Finally, we give the theoretical justification of the benefits of having the proposed explanation loss to the generalizability power of the backbone DNN model.

1.2 Contribution

The major contributions of the research presented here can be stated as follows:

Interpretable and Efficient Bio-inspired Deep Learning via Neuronal Assemblies:

- **Proposing a novel bio-inspired regularization that enhance the intrinsic interpretability and efficiency of deep neural networks**. we propose a Biologically Enhanced Artificial Neuronal assembly (BEAN) regularization that promoting jointly sparse and efficient encoding of rich semantic correlation among neurons, and enhancing model generalizability with few training samples.

- **Validating the interpretability and modularity**. Modeling neural correlations and dependencies allows us to better interpret and visualize the learned representation in hidden layers at the neuron population level instead of the single neuron level. Both qualitative and quantitative analyses show that BEAN enables the formations of identifiable neuronal assembly patterns in the hidden layers, enhancing the modularity and interpretability of the DNN representations.

- **Validating the sparse and efficient encoding of rich semantic correlation among neurons**. We show that BEAN can promote jointly sparse and efficient encoding of rich semantic correlation among neurons in DNNs similar to connection patterns in BNNs. Experimental results show that BEAN not only enables the formation of neuronal functional clusters that encode rich semantic correlation, but also allows the model to achieve state-of-the-art memory/computational efficiency without loss of model performance.

Interpretation for Dynamic Attributed Graphs via Hierarchical Attention:

- **Defining the novel problem of inferring user health stage information using online health forum data.** We define the health stage inference problem in online health forums and formulate the user activities as transition graphs that are capable of modeling user dynamic transitions between subforums and their complex relationships.

- **Proposing a generic framework DynGraph2Seq for inferring target sequence from a sequence of graphs.** We propose a novel deep neural encoder-decoder framework for learning the mapping between complex dynamic graph sequence inputs and the target output sequence.

- **Proposing dynamic graph regularization and a dynamic graph hierarchical attention mechanism for enhancing model effectiveness and interpretability.** We propose a dynamic graph regularization that enforces the smooth learning of consecutive graphs while preserving the heterogeneity across the graph sequence. In addition, we propose a new dynamic graph hierarchical attention mechanism that captures both the time-level and node-level attention, thus providing model transparency throughout the whole inference process.

Explanation-Guided Representation Learning on Geometric Data:

- **Developing generic EGL frameworks for adaptively learning how to explain geometric neural networks such as CNNs and GNNs with weak explanation supervision.** We present new learning objectives for joint optimization among the model prediction loss, the explanation loss, and the explanation regularization loss on regulating the model explanation of geometric data, including image and graph-structured data. In addition, our framework can treat the explanation loss as an optional term and thus work effectively in scenarios where the human annotation on explanation is limited.

- **Developing a unified graph-based explanation framework for calculating both node-level and edge-level explanation of GNNs.** We proposed a unified framework for both node-level and edge-level explanations that is suitable for explanation supervision and generalizable to the existing differentiable explanation methods.

- **Developing a robust model objective that can handle the noisy human annotation labels as the supervision signal for CNNs.** We propose a novel robust explanation loss that can handle the inaccurate boundary, incomplete region, as well as inconsistent distribution challenges of image data in applying the noisy human annotation labels as the supervision signal.

- **Conducting comprehensive experiments to validate the effectiveness of the proposed model.** Extensive experiments on multiple real-world datasets in geometric data domains, including chemical (molecular graphs) and vision (natural images and scene graphs), demonstrate that the proposed EGL models improved the backbone DNN model both in terms of prediction power and explainability across different application domains. In addition, qualitative analyses, including case studies and user studies of the model explanation, are provided to demonstrate the effectiveness of the proposed framework.

1.3 Thesis Organization

The remainder of the research proposal is as follows. Chapter 2 describes the proposed bio-inspired regularization and its applications to sparse learning, few sample learning. Experiments results and discussions are also presented for this line of work. Chapter 3 defines the novel problem of inferring user health stage information using online health forum data, and describes the proposed DynGraph2Seq framework for inferring target sequence from a sequence of graphs. The experimental results as well as the explainability studies on the proposed dynamic graph regularization and dynamic graph hierarchical attention mechanism are also presented. Chapter 4 introduces the proposed generic Explanation-Guided Representation Learning frameworks for adaptively learning how to explain geometric neural networks including EGL on CNNs and GNNs. The related works, problem formulations, the proposed frameworks, as well as extensive experimental results on multiple real-world datasets are presented individually for image and graph-structured data. Finally, Chapter 5 summarizes the work carried out, lists the associated publications, and suggests directions for future research.

Chapter 2

Interpretable and Efficient Bio-inspired Deep Learning via Neuronal Assemblies

2.1 Introduction

Deep neural networks (DNNs) are known for extracting useful information from a large amount of data [17]. Despite the success and popularity of DNNs in a wide variety of fields, including computer vision [74, 56] and natural language processing [31, 170], there are still many drawbacks and limitations of modern DNNs, including lack of interpretability [178], the requirement of large data [68], and post selection on complex model architecture [189, 188]. Specifically, the representations learned in DNNs are typically hard to interpret, especially in dense (fully connected) layers. Despite recent attempts to build intrinsically more interpretable convolutional units [178, 123], the exploration of learned representations in the dense layer has remained limited. In fact, dense layers are the fundamental and critical component of most state-of-the-art DNNs, which are typically used for the late stage of the network's computation, akin to the inference and decision-making processes [74, 135, 56]. Thus improving the interpretability of the dense layer representation is crucial if we are to fully understand and exploit the power of DNNs.

However, interpreting the representations learned in dense layers of DNNs is typically a very challenging task. One crucial issue of the classical DNN model such as multilayer perceptron (MLP) is that neurons in the same layer of DNNs are conditionally independent of each other, as dense layers in MLP are typically activated by all-to-all feed-forward neuron activity and trained by all-to-all feedback weight adjustment. In this comprehensively 'vertical' connectivity, every node is independent and abstracted 'out of the context' of the other nodes. This issue limits the analysis of the representation learned in DNNs to single-unit level, as opposed to the higher modularity in principle afforded by neuron population coding. Moreover, recent studies on single unit importance seem to suggest that individually selective units may have little correlation with overall network performance [101, 192]. Specifically, [101, 192] conducted unit-level ablation experiments on CNNs trained on large scale image datasets and found that ablating any individual unit does not hurt overall classification accuracy.

On the other hand, understanding the complex patterns of neuron correlations in biological neural networks (BNNs) has long been a subject of intense interest for neuroscience researchers. Circuitry blueprints in the real brain are 'filtered' by the physical requirements of axonal projections and the consequent need to minimize cable while maximizing connections. One could naively expect that the non-all-to-all limitations imposed in natural neural systems would be detrimental to their computational power. Instead, it makes them superiorly efficient and allows cell assemblies to emerge. Neuronal assemblies or cell assemblies [57] can be described as groups of neurons interconnected by strong synaptic interactions and sharing joint semantic content. The resulting population coding is essential for human cognitive and mnemonic processes [19].

In this paper, we bridge such a crucial gap between DNNs and BNNs by modeling the neuron correlations within each layer of DNNs. Leveraging biologically inspired learning rules in neuroscience and graph theory, we propose a novel Biologically-Enhanced Artificial Neuronal assembly (BEAN) regularization that can enforce dependencies among neurons in dense layers of DNNs without substantially altering the conventional architecture. The resultant advantages are threefold:

- **Enhancing interpretability and modularity at the neuron population level.** Modeling neural correlations and dependencies allows us to better interpret and visualize the learned representation in hidden layers at the neuron population level instead of the single neuron level. Both qualitative and quantitative analyses show that BEAN enables the formations of identifiable neuronal assembly patterns in the hidden layers, enhancing the modularity and interpretability of the DNN representations.

- **Promoting jointly sparse and efficient encoding of rich semantic correlation among neurons.** Here, we show that BEAN can promote jointly sparse and efficient encoding of rich semantic correlation among neurons in DNNs similar to connection patterns in BNNs. BEAN enables the model to parsimoniously leverage available neurons and possible connections through modeling structural correlation, yielding both connection-level and neuron-level sparsity in the dense layers. Experimental results show that BEAN not only enables the formation of neuronal functional clusters that encode rich semantic correlation, but also allows the model to achieve state-of-the-art memory/computational efficiency without loss of model performance.

- **Improving model generalizability with few training samples.** Humans and animals can learn and generalize to new concepts with just a few trials of learning, while DNNs generally perform poorly on such tasks. Current few-shot learning techniques in deep learning still rely heavily on a large amount of additional knowledge to work well. For example, transfer-learning-based methods typically leverage a model pre-trained with a large amount of data [161, 139], and meta-learning-based methods require a large number of additional side tasks [41, 137]. Here we explore BEAN with a substantially more challenging *few-shot learning from scratch* task first studied by [68], where no additional knowledge is provided aside from a few training observations. Extensive experiments show that BEAN has a significant advantage in improving model generalizability over conventional techniques.

2.2 Biologically-Enhanced Artificial Neuronal Assembly Regularization

This section describes the overall objective of Biologically-Enhanced Artificial Neuronal Assembly (BEAN) regularization as well as the implementation of BEAN on DNNs, as *Layer-wise Neuron Correlation and Co-activation Divergence* to model the implicit dependencies between neurons within the same layer.

2.2.1 Layer-wise Neuron Co-activation Divergence

Due to the physical restrictions imposed by dendrites and axons [117] and for energy efficiency, biological neural systems are "parsimonious" and can only afford to form a limited number of connections between neurons. The neuron connectivity patterns of BNNs are intertwined with their activation patterns based on the principle of "*Cells that fire together wire together*", which is known as **cell assembly theory**. It explains and relates to several characteristics and advantages of BNN architecture such as modularity [107], efficiency, and generalizability, that are just the aspects in which the current DNNs are usually struggling [78]. To take advantage of the beneficial architectural features in BNNs and overcome the existing drawbacks of DNNs, we propose the Biologically-Enhanced Artificial Neuronal assembly (BEAN) regularization. BEAN ensures neurons which "wire" together with a high outgoing weight correlation also "fire" together with small divergence in terms of their activation patterns.

An example of the artificial neuronal assembly achieved by our method can be seen in Figure 2.1(d). The regularization is formulated as follows:

$$L_c^{(l)} = 1/(SN_l^2) \sum\nolimits_s \sum\nolimits_i \sum\nolimits_j A_{i,j}^{(l)} \times d(H_{s,i}^{(l)}, H_{s,j}^{(l)}) \tag{2.1}$$

where L_c is the regularization loss; the term $A_{i,j}^{(l)}$ characterizes the wiring strength (the higher value, the stronger connection) between two neurons i and j within layer l; the term $d(H_{s,i}^{(l)}, H_{s,j}^{(l)})$ models the divergence of firing patterns (the higher value, the more different the firing) between two neurons i and j on input sample s. Thus, by multiplying these two functions, we penalize those neurons with strong connectivity but high activation divergence, in line with the principles of cell assembly theory. S is the total number of input samples while N_l is the total number of hidden neurons in layer l.

Specifically, $A_{i,j}^{(l)}$ defines the connectivity relation among neuron i and neuron j in DNN, which is instantiated by our newly proposed "Layer-wise Neuron Correlation" and will be elaborated in Sections 2.2.2 and 2.2.3. On the other hand, to model the "co-firing" correlation, $d(H_{s,i}^{(l)}, H_{s,j}^{(l)})$ is defined as "Layer-wise Neuron Co-activation Divergence" which denotes the difference in the activation patterns in lth layer between $H_{s,i}^{(l)}$ and $H_{s,j}^{(l)}$ of neuron i and neuron j, respectively. Here $H_{s,i}^{(l)}$ represents the activation of neuron i in layer l for a given input sample s. The function $d(x, y)$ can be a common divergence metric such as absolute difference or square difference. In this study, we show the results for a square difference in the Experimental Study Section; the absolute difference results follow a similar trend.

Model Training: The general objective function of training a DNN model along with the proposed regularization on fully connected layer l can be written as: $L = L_{DNN} + \alpha L_c^{(l)}$, where L_{DNN} represents the general deep learning model training loss and the hyper-parameter α controls the relative strength of the regularization.

Equation 2.1 can be optimized with backpropagation [121] using the chain rule:

$$\frac{\partial L_c^{(l)}}{\partial W^{(l+1)}} = \frac{\partial A^{(l)}}{\partial W^{(l+1)}} D^{(l)}, \ \frac{\partial L_c^{(l)}}{\partial W^{(l)}} = A^{(l)} \frac{\partial D^{(l)}}{\partial H^{(l)}} \frac{\partial H^{(l)}}{\partial W^{(l)}}, \ \cdots \tag{2.2}$$

where $D^{(l)} \in \mathbb{R}^{S \times N_l \times N_l}$ of which each element is $D_{s,i,j}^{(l)} = d(H_{s,i}^{(l)}, H_{s,j}^{(l)})$.

Remark 1. *BEAN regularization has several strengths. First, it enforces interpretable neuronal assemblies without the need to introduce sophisticated handcrafted designs into the architecture, which is justified later in Section 3.1. In addition, modeling the neuron correlations and dependencies further results in sparse and efficient connectivity in dense layers, which substantially reduced the computation/memory cost of the model, as shown in Section 3.2. Besides, the encoding of rich semantic correlation among neurons may improve the generalizability of the model when insufficient data and knowledge are provided, which is demonstrated later in Section 3.3. Finally, the Layer-wise Neuron Correlation can be efficiently computed with matrix operations, as per Equations 2.5 and 2.7, which enables modern GPUs to boost up the speed during model training. In practice, we observe negligible run time overhead of the addition computation needed for BEAN regularization.*

2.2.2 The First-Order Layer-wise Neuron Correlation

This section introduces the formulation of the layer-wise neuron correlation $A_{i,j}^{(l)}$ between any pair of neurons i and j.

In the human brain, the correlation between two neurons depends on the wiring between them [20] and hence is typically treated as a binary value in BNN studies, with "1" indicating the presence of a connection and "0" the absence, so the corre-

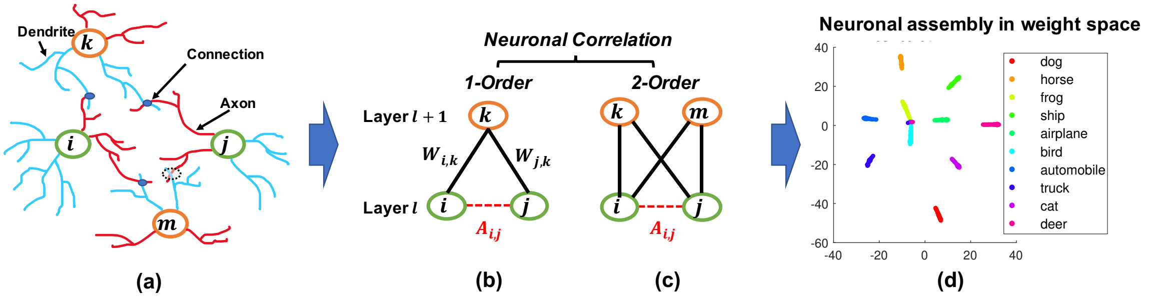


Figure 2.1: An illustration of how the proposed constraint drew inspiration from BNNs and bipartite graphs. **(a)** neuron correlations in BNNs correspond to connections between dendrites, which are represented by blue lines, and axons, which are represented by red lines. **(b)** and **(c)** analogy of figure (a) represented as connections between layers in DNNs; although nodes i and j cannot form direct links, they can be correlated by a given node k as a first-order correlation, or by two nodes k and m as a second-order correlation which is also equivalent to a 4-cycle in bipartite graphs. **(d)** an example of a learned neuronal assembly in neurons outgoing weight space, with the dimensionality reduced to 2D with T-SNE [92]. Each point represents one neuron and the neurons are colored according to their highest activated class in the test data.

lation among a group of neurons can be represented by the corresponding adjacency matrix. Although there is typically no direct connection between neurons within the same layer of DNNs, it is possible to model neuron correlations based on their connectivity patterns to the next layer. This resembles a common approach in network science, where it is useful to consider the relationships between nodes based on their common neighbors in addition to their direct connections. One classic concept widely used to describe such a pattern is called *triadic closure* [53]. As shown in Figure 2.1 (b), triadic closure can be interpreted here as a property among three nodes i, j, and k, such that if connections exist between $i-k$ and $j-k$, there is also a connection between $i-j$.

We take this scheme a step further to model the correlations between neurons within the same layer by their connections to the neurons in the next layer. This can be considered loosely analogous to the degree of similarity of the axonal connection pattern of biological neurons in BNNs [114]. To simulate the relative strength of such connections in DNNs, we introduce a function $f(\cdot)$ that converts the actual weights into a relative connectivity strength. Suppose matrix $W^{(l+1)} \in \mathbb{R}^{N_l \times N_{l+1}}$ represents all the weights between neurons in layers l and $l+1$ in DNNs, where N_l and N_{l+1} represent the numbers of neurons, respectively. The relative connectivity strength can be estimated by the following equation[1]:

$$f(W^{(l+1)}) = |tanh(\gamma W^{(l+1)})| \tag{2.3}$$

where $|\cdot|$ represents the element-wise absolute operator; $tanh(\cdot)$ represents the element-wise hyperbolic tangent function; and γ is a scalar that controls the curvature of the hyperbolic tangent function. The values of $f(W^{(l+1)}) \in \mathbb{R}^{N_l \times N_{l+1}}$ will all be positive and in the range of $[0, 1)$ with the value simulating the relative connectivity strength of the corresponding synapse between neurons.

Although there can be positive and negative weights in DNNs, our assumption on connection strength follows the typical way of BNN studies, which measures the presence and absence of the connection as mentioned above. Moreover, since DNNs require continuous values instead of discrete values to make the function differentiable for optimization, we further use Equation (2.3) to convert the concept of the presence/absence of the connections to the relative strength of the connections. More specifically, the difference is that instead of treating connection to be either “1” (indicating the presence of a connection) or “0” (indicating the absence of the connection), we treat the output of Equation (2.3) as the strength of that connection, where high values (i.e. close to “1”) indicate the presence of a strong connection and low values (i.e. close to “0”) indicate weak or no connection.

Based on this, we can now give the definition for the *layer-wise first-order neuron correlation* as:

Definition 1. Layer-wise first-order neuron correlation. For a given neuron i

[1] Similar to the ReLU activation function, our formulation introduces a non-differentiable point at zero; we follow the conventional setting by using the sub-gradient for model optimization.

and neuron j in layer l, the layer-wise first-order neuron correlation is given by:

$$A_{i,j}^{(l)} = (1/N_{l+1}) \sum\nolimits_{k=1}^{N_{l+1}} f(W_{i,k}^{(l+1)}) \times f(W_{j,k}^{(l+1)}) \tag{2.4}$$

The above formula can be expressed as the product of two matrices:

$$A^{(l)} = (1/N_{l+1}) f(W^{(l+1)}) \cdot f(W^{(l+1)})^T \tag{2.5}$$

where $\cdot$ represents the matrix multiplication operator.

The layer-wise neuron correlation matrix $A^{(l)}$ is a symmetric square matrix that models all the pairwise correlations of neurons with respect to their corresponding outgoing weights in layer l. Each entry $A_{i,j}^{(l)}$ takes a value in the range $[0, 1)$ and models the correlation between neuron i and neuron j in terms of the similarity of their connectivity patterns. The higher the value, the stronger the correlation between the two.

In this setting, two neurons i and j from layer l will be linked and correlated by an intermediate node k from layer $l+1$ if and only if both edges $f(W_{i,k}^{(l+1)})$ and $f(W_{j,k}^{(l+1)})$ are non zero, and the relative strength can be estimated by $f(W_{i,k}^{(l+1)}) \times f(W_{j,k}^{(l+1)})$, which will be in the range $[0, 1)$. Since there are N_{l+1} neurons in layer $l + 1$, where each neuron k can contribute to such connections, running over all neurons in layer $l + 1$ we obtain Equation 2.4 and Equation 2.5.

2.2.3 The Second-Order Layer-wise Neuron Correlation

Although the first-order correlation is able to estimate the degree of dependency between each pair of neurons, it may not be sufficient to strictly reflect the degree of grouping or assembly of the neurons. Thus, here we further propose a second-order neuron correlation based on the first-order correlation defined in Equation 2.4 and 2.5, as:

Definition 2. Layer-wise second-order neuron correlation. For a given neuron i and neuron j in layer l, the layer-wise second-order neuron correlation is given by:

$$A_{i,j}^{(l)} = (1/N_{l+1}^2) \sum\nolimits_{k,m} f(W_{i,k}^{(l+1)}) \times f(W_{j,k}^{(l+1)}) \times f(W_{i,m}^{(l+1)}) \times f(W_{j,m}^{(l+1)}) \qquad (2.6)$$

The above formula can be expressed as the product of four matrices:

$$A^{(l)} = (1/N_{l+1}^2)(f(W^{(l+1)}) \cdot f(W^{(l+1)})^T) \odot (f(W^{(l+1)}) \cdot f(W^{(l+1)})^T) \qquad (2.7)$$

where $\odot$ represents the element-wise multiplication of matrices.

The second-order correlation provides a stricter criterion for relating neurons, as it requires at least two common neighbor nodes from the layer above to have strong connectivity, as compared to the first-order correlation that requires just one common neighbor. Moreover, the second-order neuron correlation is closely related both to graph theory concepts and a neuroscience-inspired learning rule:

Remark 2. ***Graph theory and neuroscience interpretation.*** *Modeling the first-order correlation between two neurons within the same layer is based on the co-connection to a common neighbor neuron from the layer above, which is closely related to the concepts of clustering coefficient [156] and transitivity [60] in graph theory. On the other hand, modeling the second-order correlation between two neurons involves two common neighbor neurons in the layer above, which is closely related to calculating the 4-cycle pattern where all 4 possible connections in between are taken into account, as shown in Figure 2.1 (b). This 4-cycle pattern is linked to the global clustering coefficients of bipartite networks [118], where the set of vertices can be decomposed into two disjoint sets such that no two vertices within the same set are adjacent. Similarly, if we consider neurons within one layer as the nodes that belong to one set of the bipartite network between two adjacent layers of the neural networks, forming this 4-cycle will tend to increase the clustering coefficients of the network. Moreover, the second-order correlation is also related to several cognitive neuroscience* *studies, such as the BIG-ADO learning rule and the principal semantic components of language [94, 126] as well as the notion of discrete neuronal circuits [110]. Figure 2.1 (a) illustrates a scenario of the BIG-ADO learning rule in BNNs. The blue blobs represents a connection that was formed between two neurons (i.e., a synapse), while the dashed circle between neurons j and m represents an Axo-Dendritic Overlap (ADO) (i.e., a potential synapse) between the two neurons. BIG-ADO posits that in order to form a synapse, there must be a potential synapse in place, and the probability of having a potential synapse grows with the second-order correlation. Notably, both of the neuroscience papers cited above relate such a learning mechanism to the formation of cell assemblies in the brain, which parallels our observation of neuronal functional clusters among neurons in DNNs when BEAN was imposed, as shown in Figure 2.1 (c) and Figure 2.6 (b).*

2.3 Experimental Study

Our description of the empirical analysis design and results is organized in the following fashion. In Section 3.1, we first characterize the interpretable patterns from the learning outcomes of BEAN regularization on multiple classic image recognition tasks. We then further analyze in Section 3.2 how BEAN could benefit the model from learning sparse and efficient neuron connections. Finally, in Section 3.3 we study the effect of BEAN regularization on improving the generalizability of the model on several few-shot learning from scratch task simulations. We refer to both distinct BEAN variations, BEAN-1 and BEAN-2, based on the two proposed layer-wise neuron correlation defined by Equation 2.5, and Equation 2.7 respectively. The value for γ (Equation 2.3) was set to 1. This paper focuses on examining the effects of the proposed regularization rather than the differences between distinct types of neural network architectures. Hence, we simply adopted several of the most popular neural network architectures for the chosen datasets and did not perform any hyperparameter or system parameter tuning using the test set; in other words, we did not perform any ”post selection” (i.e. selectively reporting the model results based on testing set [188, 189]). All network architectures used in this paper are fully described in their respective cited references, including the specification of their system parameters. The regularization factor of BEAN and other baseline methods were chosen based on the model performance on the validation set. All the experiments were conducted on a 64-bit machine with Intel(R) Xeon(R) W-2155 CPU 3.30GHz processor and 32GB memory and an NVIDIA TITAN Xp GPU.

2.3.1 The Interpretable Patterns of BEAN Regularization

Due to the highly complex computation among numerous layers of neurons in traditional DNNs, it is typically difficult to understand how the network learned what it remembers and the system is more commonly treated as a black-box model [179]. Here, to ascertain the effect of BEAN regularization on the interpretability of network dynamics, we analyze the differences in neuronal representation properties of the DNNs with and without BEAN regularization. We conducted experiments on three classic image recognition tasks on the MNIST [77], Fashion-MNIST [162] and CIFAR-10 [73] datasets by starting with three predefined network architectures as listed below:

1. An MLP with one hidden layer of 500 neurons with ReLU activation function for MNIST and Fashion-MNIST datasets.

2. A LeNet-5 [77] for MNIST and Fashion-MNIST datasets.

3. ResNet18 [56] for CIFAR-10 dataset.

The Adam optimizer [69] was used with a learning rate of 0.0005 and a batch size of 100 for model training until train loss convergence was achieved; BEAN was applied to all the dense layers of each model.

Biological plausibility of the learned neuronal assemblies

By analyzing the neurons' connectivity patterns based on their outgoing weights, we discovered neuronal assemblies in dense layers where BEAN regularization was enforced. Specifically, for both datasets, we found that the neuronal assemblies at the last dense layer could be best described by 10 clusters with K-means clustering [93] validated by Silhouette analysis [120]. Silhouette analysis is a widely-used method for interpretation and validation of consistency within clusters of data. The technique provides a succinct graphical representation of how well each object has been classified. As shown in Figure 2.2, we visualized the K-means clustering results in neurons' weight space of the dense layer on both MNIST (top) and CIFAR-10 (bottom) datasets. Each data point in the figure indicates one single neuron and the color indicates its cluster assignment by the clustering algorithm. The Silhouette value is further used to assess the quality of the clustering assignment: high Silhouette values support the existence of clear clusters in the data points, which here correspond to neural assembly patterns among neurons.

Both BEAN-1 and BEAN-2 could enforce neuronal assemblies for various models on several datasets, yielding Silhouette indices around 0.9, which indicates strong clustering patterns among neurons in dense layers where BEAN regularization was applied. On the other hand, training conventional DNN models with the same architectures could only yield Silhouette indices near 0.5, which indicates no clear clustering patterns in conventional dense layers of deep neuronal networks.

Moreover, we found co-activation behavior of neurons within each neuronal assembly, which is both interpretable and biologically plausible. Figure 2.3 shows the visualization of neuron co-activation patterns found in the last dense layer of LeNet-5+BEAN-2 model on MNIST dataset. For the samples of each specific class, only

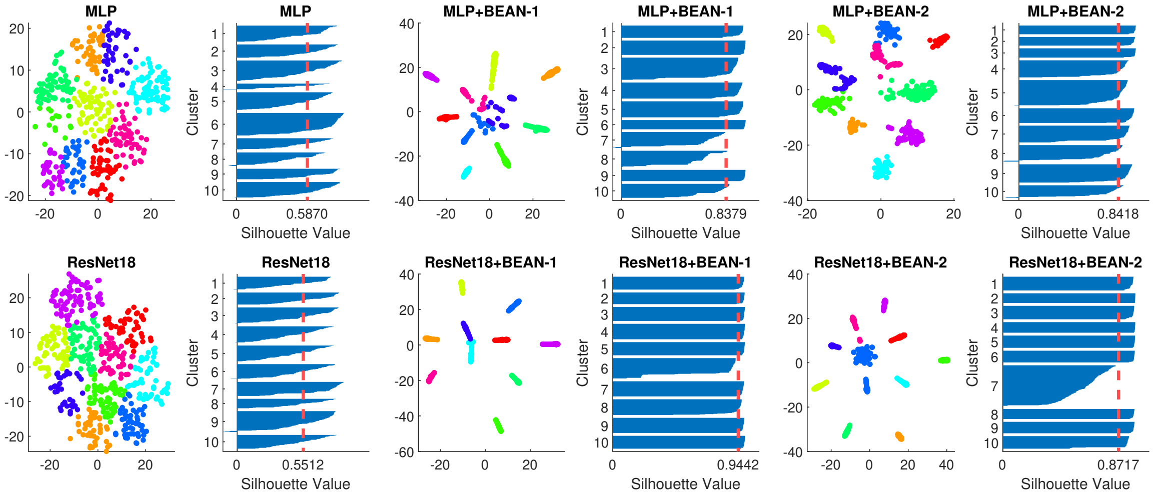


Figure 2.2: Neuronal assembly patterns found in neurons' weight space of the dense layer of different models on both MNIST (top) and CIFAR-10 (bottom) datasets, along with clustering validation via Silhouette score on 10 clusters K-means clustering. The dimensionality of neurons' weight space was reduced to 2D with T-SNE for visualization.

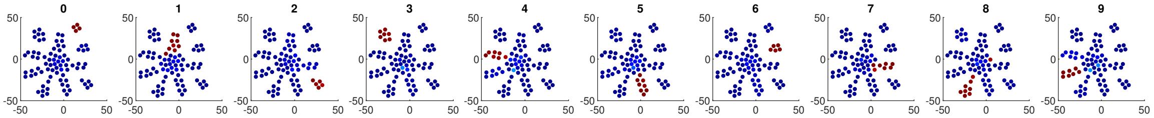


Figure 2.3: Neuron co-activation patterns found in the representation of the last dense layer of LeNet-5+BEAN-2 model on MNIST dataset. The dimensionality of neurons' weight space was reduced to 2D with T-SNE for visualization. Each point represents one neuron within the last dense layer of the model and is colored based on its activation scale. The 10 subplots show the average activation heat-maps when each digit's samples were fed into the model. The warmer color indicates a higher neuron activation.

those neurons in the specific neuron group that is associated with that digit class have high activation while all the other neurons remain silent. This strong correlation between each unique assembly and each unique class concept allows straightforward interpretation of the neuron populations in the dense layers. From the neuroscience perspective, those co-activation patterns and the association between high-level concepts and neuron groups may reflect similar co-firing patterns observed in biological neural systems [107] and underscore the strong association between neuronal assembly and concepts [146] in biological neural networks.

We also found a strong correlation between neuronal assembly and class selectivity indices. Selectivity index was originally proposed and used in systems neuroscience [34, 42]. Recently, machine learning researchers also studied unit class selectivity [101, 192] as a metric for interpreting the behaviors of single units in deep neural networks. Mathematically, it is calculated as: $selectivity = (\mu_{max} - \mu_{-max})/(\mu_{max} + \mu_{-max})$, where μ_{max} represents the highest class-conditional mean activity and μ_{-max} represents the mean activity across all other classes.

To better visualize how high-level concepts are associated with the learned neuron assemblies, we further labeled each neuron with the class in which it achieved its highest class-conditional mean activity μ_{max} in the test data. Figure 2.4 shows the results for the last dense layer of the models trained with both datasets. We found that the neuronal assembly could be well described based on selectivity. The strong association between neuronal assemblies and neurons' selectivity index further demonstrated the biological plausibility of the learning outcomes of BEAN regularization. Moreover, the strong neuron activation patterns towards each individual high-level concepts or classes could in principle enable one to better understand what each individual neuron has learned to represent. However, more relevant to and consistent with our regularization, these selective activation patterns reveal how a group of neurons (i.e. neuronal assembly) together capture the whole picture of each high-level concept,

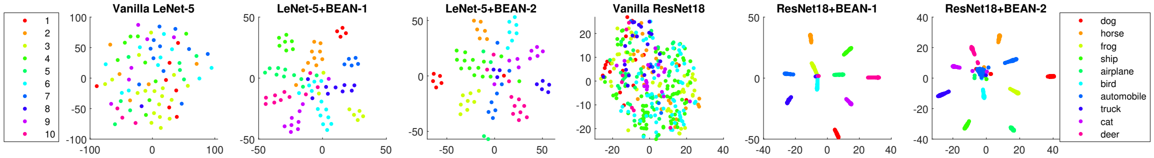


Figure 2.4: The strong association between neuronal assemblies and neurons' class selectivity index with BEAN regularization on both MNIST (left) and CIFAR-10 (right) datasets. Each point represents one neuron and the color represents the class where the neuron achieved its highest class-conditional mean activity in the test data.

such as the 'bird' class in CIFAR-10 as shown in Figure 2.4.

In this subsection, we have demonstrated the promising effect of the proposed BEAN regularization on forming the neural assembly patterns among the neurons in the last layer of the network and their correspondence with biological neural networks. Although the effect of BEAN regularization is not yet clear on the lower layers of the networks, it will be interesting in the future to explore additional relations between computational function and the architecture of earlier processing stations in biological neural systems.

Quantitative analysis of interpretability

Experimental neuropsychologists commonly use an ablation protocol when studying neural function, whereas parts of the brain are removed to investigate the cognitive effects. Similar ablation studies have also been adapted for interpreting deep neural networks, such as understanding which layers or units are critical for model performance [50, 101, 192].

To quantitatively evaluate and compare interpretability, we performed an ablation study at the neuron population level, each time ablating one distinct group of neurons and recording the consequent model performance changes for each class. As shown in Figure 2.4, we identified neuron groups via class selectivity and performed neuron population ablation accordingly. Figure 2.5 shows the results of all 10 ablation runs for each class in MNIST dataset. As also reported by [101], for conventional deep neural

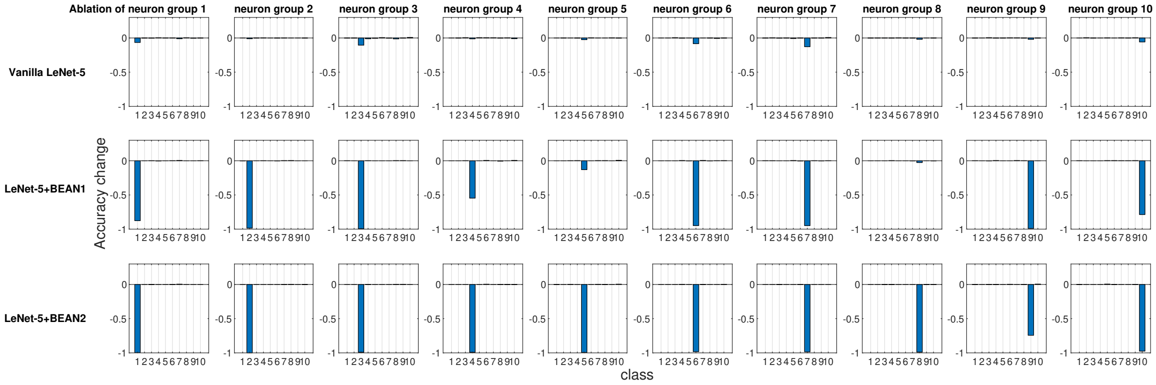


Figure 2.5: The ablation study at the neuron population level of the last dense layer of LeNet-5 models. Each time, one distinct group of neurons were ablated based on their most selective class and the model performance changes for each individual class were recorded.

nets, there is indeed no clear association between neuron's selectivity and importance to the overall model performance, as revealed by neuron population ablation. However, when BEAN regularization was utilized during training, such association clearly emerged, especially for BEAN-2. This is because BEAN-2 could enforce neurons to form stricter neuron correlations than BEAN-1 with the second-order correlation, enabling groups of neurons to represent more compact and disentangled concepts, such as handwritten digits. This discovery further demonstrated the interpretability and concept level representation in each neuronal assembly learned by applying BEAN regularization. Such compact and interpretable structure of concept-level information encoding could also benefit the field of disentanglement representation learning [17].

2.3.2 Learning Sparse and Efficient Networks

To evaluate the effect of BEAN regularization on learning sparse and efficient networks, we conducted experiments on two real-world benchmark datasets, i.e., the MNIST [77] and Fashion-MNIST [162] datasets. We compared BEAN with several state-of-the-art regularization methods that could enforce sparse connection of the network, including ℓ_1-norm, group sparsity based on $\ell_{2,1}$-norm [175, 6], and exclusive sparsity based on $\ell_{1,2}$-norm [194, 71]. Notable studies also investigated the combination of the sparsity terms listed above, such as combining group sparsity and ℓ_1-norm [127], and combining group and exclusive sparsity [169]. The combinatorial study is outside the scope of this work, as our focus is on showing and comparing the effectiveness of the single regularization term to the network. To keep the comparison fair and accurate, we use the same base network architecture for all regularization methods tested in this experiment, which is a predefined fully connected neural network with 3 hidden layers, 500 neurons per layer, and ReLU as the neuron activation function. The regularization methods are applied to all layers of the network, except the bias term. The regularization co-efficients are selected through a grid search varying from 10^{-5} to 10^3 based on the model performance on the validation set, as shown in Algorithm 1. To obtain a more reliable and fair result, we ran a total of 20 random weight initializations for every network architecture studied and reported the overall average performance of all 20 results as the final model performance of each architecture.

Algorithm 1: The pseudo code for searching for the best α value in BEAN

```
func hyperparameter_tuner(training_data, validation_data, alpha_list =
  $[0.001, 0.01, 0.1, 1, 10, 100]$) :
    hp_perf = []
    % train and evaluate on all hyper-parameter settings
    foreach $\alpha$ in alpha_list :
        $m$ = train_model(training_data, alpha)
        validation_results = eval_model($m$, validation_data)
        hp_perf.append(validation_results)
    % find the best alpha on validation set
    best_alpha = alpha_list[max_index(hp_perf)]
    return best_alpha
```

To quantitatively measure the performance of various sparse regularization techniques, we used three evaluation metrics, including the prediction accuracy on test data (i.e. measured by the number of correct predictions divided by the total number of samples in test data), the ratio of parameters used in the network (i.e. total number of non-zero weights divided by the total number of weights in the networks after training), and the corresponding number of floating point operations (FLOPs). A higher accuracy means that the model can train a better network for the classification tasks. A lower FLOP indicates that the network needs fewer computational operations per forwarding pass, which reflects computation efficiency. Similarly, a lower parameter usage indicates the network requires less memory usage, which reflects memory efficiency.

Table 2.1: Efficient model learning experiments on MNIST and Fashion-MNIST datasets. The FLOPs and effective parameters (i.e. number of non-zero parameters) are normalized by the value of vanilla model. Performance is averaged over 20 runs. The best and second-best results are highlighted in boldface and italic font, respectively.

Dataset	Measure	Vanilla	ℓ_1-norm	Group Sparsity	Exclusive Sparsity	BEAN-1	BEAN-2
MNIST	accuracy	0.9812	*0.9835*	0.9813	0.9824	**0.9842**	0.9823
	FLOPs	1	0.8106	0.6098	0.4248	*0.2212*	**0.1320**
	parameter	1	0.2921	*0.0982*	0.1375	0.1496	**0.0730**
Fashion-MNIST	accuracy	**0.8986**	0.8924	0.8925	0.8930	*0.8960*	0.8916
	FLOPs	1	0.8011	0.5384	0.5320	*0.2913*	**0.1622**
	parameter	1	0.4357	*0.1378*	0.2257	0.2592	**0.1259**

The results are shown in Table 2.1. For each evaluation metric, the best and second-best results are highlighted in boldface and italic font, respectively. As can be seen, both BEAN-1 and BEAN-2 can achieve high memory and computational efficiency without sacrificing network performance for the classification tasks. Specifically, BEAN-2 achieved the best memory and computational efficiency, out-performing baseline models by 25-75% on memory efficiency and 69-84% on computational efficiency on the MNIST dataset, and by 9-71% on memory efficiency and 69-80% on computational efficiency on the Fashion-MNIST dataset. BEAN-1 also achieved a good trade-off between model performance and efficiency, being the second-best on computational efficiency and the best on model performance on both the MNIST and Fashion-MNIST datasets. Comparing with BEAN-2, BEAN-1 leans more toward the model performance side in such a trade-off. This is because the first-order correlation used in BEAN-1 is less restrictive than a higher-order correlation in BEAN-2, as only one support neuron in the layer above is enough to build up a strong correlation. Thus, in practice, using a higher-order correlation might be promising when the objective is to learn a more efficient model.

Interestingly, BEAN regularization seemed to advance the state-of-the-art by an even more significant margin in terms of computational efficiency. In fact, BEAN regularization reduces the number of FLOPs needed for the network by automatically "pruning" a substantial proportion of neurons in the hidden layers (whereas a neuron is considered pruned if either all incoming or all outgoing weights are zero), due to the penalization of connections between neurons that encode divergent information. Although group sparsity and exclusive sparsity are designed to achieve a similar objective for obtaining neuron-level sparsity, they are less effective than BEAN regularization. This is due to the fact that BEAN takes into consideration not only the correlations between neurons via their connection patterns but also the consistency of those correlations with their activation patterns.

We have shown in Table 2.1 that the proposed BEAN regularization can effectively make the connection sparser in the dense layers of the artificial neural networks. In general, this 'sparsifying' effect can be beneficial for any models with at least one dense layer in the network architecture. Most modern deep neural networks (such as VGG [135] and ImageNet [122]) can enjoy this sparsity benefit, as the dense layers typically contribute to the majority of the model parameters [23].

2.3.3 Towards few-shot learning from scratch with BEAN regularization

In an attempt to test the influence of BEAN regularization on the generalizability of DNNs in the scenarios where the training samples are extremely limited, we conducted a *few-shot learning from scratch* task, i.e. without the help of any additional side tasks and pre-trained models [68]. Notice that in the few-shot learning setting, the model typically requires an iterative learning process over the sample set. In other words, for each individual few-shot learning experiment, only a few image samples per digit are randomly selected to form the training set. The model then iteratively learns from the selected image samples until convergence is achieved. So far, this kind of learning task has rarely been explored due to the difficulty of the problem setup as compared to other conventional few-shot learning tasks where additional data or knowledge could be accessed. Currently, only [68] carried out a preliminary exploration with their proposed Imitation Networks model. We conducted several simulations of the *few-shot learning from scratch* task on the MNIST [77], Fashion-MNIST [162], and CIFAR-10 [73] datasets. Besides Kimura's Imitation Networks, we also compared BEAN with other conventional regularization techniques commonly used in the deep learning literature. Specifically, we compared dropout [141], weight decay [75], and ℓ_1-norm. Similarly to the description of Section 2.3.2, we kept the comparison fair and accurate by using a predefined network architecture, namely LeNet-5 [77], as the base network architecture for all regularization methods studied in this experiment. The regularization terms were applied to all three dense layers of the base LeNet-5 network. Once again, the hyperparameter of each regularization along with all other system parameters were selected through a grid search and based on the best performance on a predefined 10k validation set sampled from the original training base and completely distinct from the training samples used in the few-shot learning tasks and the testing set.

Table 2.2 shows model performance on several *few-shot learning from scratch* experiments on the MNIST, Fashion-MNIST, and CIFAR-10 datasets. Performance is averaged over 20 experiments of randomly sampled training data from the original training base. The best and second-best results for each few-shot learning settings are highlighted in boldface and italic font, respectively. As can be seen, the proposed BEAN regularization advanced the state-of-the-art by a significant margin on

Table 2.2: Few-shot learning from scratch experiments on the MNIST (left), Fashion-MNIST (middle), and CIFAR-10 (right) datasets. Performance is averaged over 20 simulations of randomly sampled training data from the original training base. The best and second-best results for each few-shot learning setting are highlighted in boldface and italic font, respectively.

Dataset	MNIST				Fashion-MNIST				CIFAR-10			
Model	1-shot	5-shot	10-shot	20-shot	1-shot	5-shot	10-shot	20-shot	1-shot	5-shot	10-shot	20-shot
Vanilla	38.63	70.21	78.97	86.68	39.32	59.02	64.50	70.23	15.60	18.49	22.45	26.39
Dropout	40.13	72.45	82.04	89.22	40.78	60.04	65.40	71.83	15.10	18.85	22.73	26.01
Weight decay	39.51	71.76	82.87	90.15	41.31	61.98	67.25	71.88	15.47	19.17	23.74	26.77
ℓ_1-norm	40.96	74.35	81.17	90.68	41.26	62.18	67.30	70.85	15.64	18.95	23.16	26.99
Imitation networks	44.10	70.40	80.00	86.70	44.80	62.10	68.00	72.50	-			
BEAN-cos	*54.05*	80.16	86.28	92.22	42.48	65.49	68.97	74.20	18.23	*21.45*	24.66	28.74
BEAN-1	**54.79**	**83.42**	*87.51*	*92.79*	**50.57**	**66.95**	*69.21*	*74.25*	**19.39**	**21.92**	*24.81*	*28.95*
BEAN-2	53.75	*80.76*	**88.08**	**92.97**	*49.94*	*65.98*	**70.21**	**75.06**	*19.28*	21.28	**25.04**	**29.23**

all four *few-shot learning from scratch* tasks tested among all three datasets. Moreover, BEAN advanced the performance more significantly when training samples were more limited. For instance, BEAN outperformed all comparison methods by 24-42%, 13-29%, and 24-28% on 1-shot learning tasks on the MNIST, Fashion-MNIST, and CIFAR-10 datasets, respectively. This observation demonstrates the promising effect of BEAN regularization on improving the generalizability of the neural nets when the training samples are extremely limited. Another interesting observation is that BEAN-1 in general performed the best with extremely limited training samples, such as the 1-shot and 5-shot learning tasks, while BEAN-2 regularization in general performed the best with slightly more training samples, such as the 10-shot and 20-shot learning tasks. The reason behind this observation might be related to the more stringent higher-order correlation, which requires more common neighbor neurons that appear to have strong connections with both neurons. Thus, a modestly increased availability of sample observations could enable BEAN-2 to form more effective neuronal assemblies, further improving the model performance.

Furthermore, we studied an additional variant for BEAN, i.e. BEAN-cos, which calculates the layer-wise neuron correlation via cosine similarity between the downstream weights of two neurons. As shown in Table 2.2, we found that BEAN-cos can still yield good performance and beat other existing regularization methods, and getting competitive results as compared with BEAN-1 and BEAN-2. However, it is inferior to BEAN-1 in 1-shot and 5-shot settings, and inferior to BEAN-2 in 10-shot and 20-shot settings. This is because BEAN-cos is unable to handle the order of correlation between neurons, as using cosine similarity requires us to treat the out-going weights of a neuron as a whole (vector) to compute the pair-wise similarity between neurons. Thus, doing this will lose the ability to calculate higher-order correlation (such as the second-order correlation), and consequentially lose the good interpretation from graph theory and neuroscience (as described in Remark 2).

To better understand why BEAN regularization could help the seemingly over-parameterized model generalize well on a small sample set, we further analyzed the learned hidden representation of the dense layers where BEAN regularization was employed. We found that BEAN helped the model gain better generalization power in two aspects: 1) by automatic sparse and structured connectivity learning and 2) by weak parameter sharing among neurons within each neuronal assembly. Both aspects enhanced the dense layers to promote efficient and parsimonious connections, which consequently prevented the model from over-fitting with a small training sample size.

Figure 2.6 shows the learned parameters of the last dense layer of LeNet-5+BEAN2 on the MNIST 10-shot learning task. As shown in Figure 2.6 (b), instead of using all possible weights in the dense layer, BEAN caused the model to parsimoniously leverage the weights and even the neurons, yielding a bio-plausible sparse and structured connectivity pattern. This is because the learned neuron correlation helped the model disentangle the co-connections between neurons from different assemblies, as shown in Figure 2.6 (a). Additionally, BEAN enhanced parameter sharing among neurons within each assembly, as demonstrated in Figure 2.6 (c). For instance, neurons in the red-colored assembly all had high positive weights toward class 4, meaning that this group of neurons was helping the model identify Digit 4. Similarly, neurons in the green-colored assembly were trying to distinguish between Digits 9 and 7. Such au-

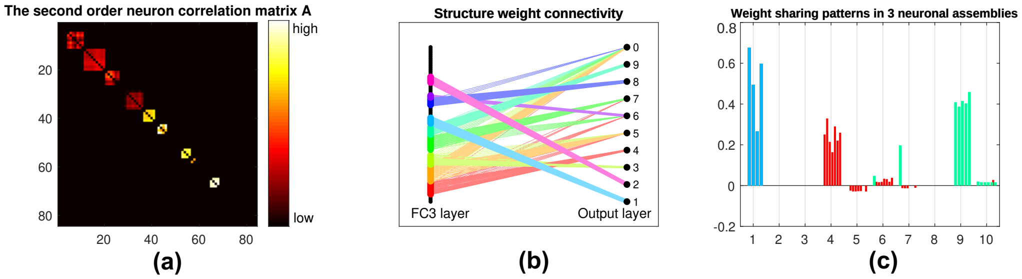


Figure 2.6: Analysis and visualization of the last dense layer of LeNet-5+BEAN-2 model on the MNIST 10-shot learning from scratch task. **(a)** Heat-map of the learned second-order neuron correlation matrix: neuron indices are re-ordered for best visualization of neuronal assembly patterns; BEAN is able to enforce plausible assembly patterns that act as functional clusters for the categorical learning task. **(b)** Visualization of the parsimonious connectivity learned in the dense layer: both neuron-level and weight-level sparsity are simultaneously promoted in the network after applying BEAN regularization. The neurons are grouped and colored by neuronal assemblies. **(c)** Visualization of the scales of neurons' outgoing weights: the weights of the neurons are colored to be consistent with the neuron group in (b).

Table 2.3: Statistic of data values test set error rate - validation set error rate on 10-shot learning on the MNIST dataset from 20 random runs. Other n-shot learning settings follow the same trend.

Model / Metric	Max	75%-rank	50%-rank	Mean	25%-rank	Min
Vanilla	0.06%	0.01%	-0.06%	-0.30%	-0.74%	-0.81%
BEAN-1($\alpha = 1$)	-0.04%	-0.20%	-0.62%	-0.58%	-0.93%	-1.13%
BEAN-2($\alpha = 100$)	0.10%	-0.02%	-0.42%	-0.48%	-0.97%	-1.35%

tomatic weak parameter sharing not only helped prevent the model from over-fitting but also enabled an intuitive interpretation of the behavior of the system as a whole from a higher modularity level.

Parameter sensitivity study

There are two hyperparameters in the proposed BEAN regularization: 1) α, which balances between the regularization loss and DNN training loss, and 2) γ, which controls the curvature of the hyperbolic tangent function as shown in Equation 2.3. As already mentioned in the first paragraph of Section 3, γ was set to 1 for all experiments. Thus, the only parameter we need to study is α.

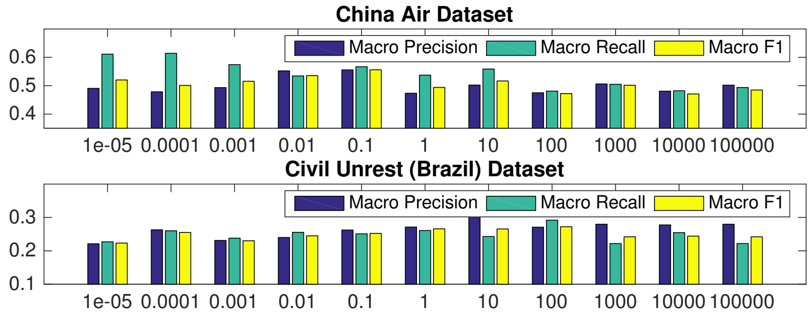


Figure 2.7: Parameter sensitivity study of BEAN regularization on 10-shot learning on the MNIST dataset. Each data point is centered by the mean value and the error bar measures the standard deviation over 20 runs.

Figure 2.7 shows the accuracy of the model versus α on the few-shot learning setting on the MNIST dataset. Only the results for the 10-shot learning task are shown due to space limitations. By varying α across the range from 0.001 to 100, the best performance is obtained when $\alpha = 1$ for BEAN-1 and $\alpha = 100$ for BEAN-2. Specifically, for BEAN-1, We can see a clear trend where the model performance drops when α is too small or too big. Furthermore, the results show that the performance of the validation set is well aligned with the model performance on the test set, this demonstrates the superior generalizability of the model when applying BEAN regularization. Notably, although in Figure 2.7 we accessed the model performance on multiple settings of α, we did not use any of the results on the test set to choose any parameters of the model, i.e. no post-selection was performed. We believe post-selection should be completely avoided and it can cause the test set to lose its power to test the model's generalizability to future unseen data.

2.4 Conclusion

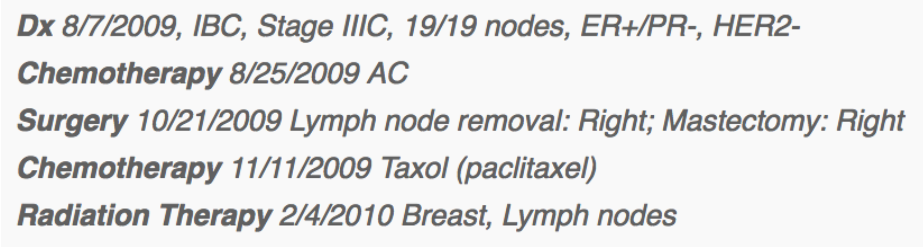
In this work, we propose a novel Biologically Enhanced Artificial Neuronal assembly (BEAN) regularization to model neuronal correlations and dependencies inspired by cell assembly theory from neuroscience. We show that BEAN can promote jointly sparse and efficient encoding of rich semantic correlation among neurons in DNNs similar to connection patterns in BNNs. Experimental results show that BEAN enables the formations of interpretable neuronal functional clusters and consequently promotes a sparse, memory/computation-efficient network without loss of model performance. Moreover, our few-shot learning experiments demonstrated that BEAN could also enhance the generalizability of the model when training samples are extremely limited. Our regularization method has demonstrated its capability in enhancing the modularity of the representations of neurons for image semantic meanings such as digits, animals, and objects on image datasets. While the generality of the approach introduced here is at this time evaluated on MNIST and CIFAR datasets, future studies might consider additional experiments on other datasets such as texts or graphs to demonstrate the broader effectiveness of the proposed method. Another direction to further enhance the model might be to include separate excitatory and inhibitory nodes, as in BNNs, which would allow implementation of specific microcircuit computational motifs [9]. Furthermore, since there are other choices for defining the affinity matrix between neurons in a certain layer based on their downstream weights, answering the question about “what is the best way to compute affinity matrix” can be an interesting direction to be more comprehensively studied in future works.

Chapter 3

Interpretation for Dynamic Attributed Graphs via Hierarchical Attention

3.1 Introduction

Online healthcare forums and communities such as the Breast Cancer Community [2], American Cancer Society [1], and eHealth Forum [3] have greatly changed the way patients seek health-related information and have become an important part of patients' lives. Unlike traditional approaches, where patients only receive information about their disease from their care providers, these online forums and communities have enabled millions of patients to ask questions related to their diseases, interact with other patients with similar prognoses, and provide support to each other across the world. The communications and interactions between patients in online forums can provide valuable information about a patient's emotional well-being and behaviors related to the management of their health that conventional clinical data collected from hospital information systems and electronic health records (EHR) is unable to capture.



Dx *8/7/2009, IBC, Stage IIIC, 19/19 nodes, ER+/PR-, HER2-*
Chemotherapy *8/25/2009 AC*
Surgery *10/21/2009 Lymph node removal: Right; Mastectomy: Right*
Chemotherapy *11/11/2009 Taxol (paclitaxel)*
Radiation Therapy *2/4/2010 Breast, Lymph nodes*

Figure 3.1: An example of patient signature that contains cancer diagnosis and treatment history.

Moreover, beyond conventional online communities and social media, online health communities provide a unique way to analyze and infer patients' health stages and disease history. Figure 3.1 shows an example of a patient's health stage information extracted from a patient signature that contains the cancer diagnosis and treatment history, along with the relevant dates. In all, the synergies between the information on patients' online communication and health status make possible a unique and wide range of research topics on health informatics, such as patient behavior statistical analyses [155, 37, 181], longitudinal communication network analyses [182, 180], and patient participation prediction [124], all of which rely on both patients' interactions in online forums as well as their health stage records.

However, the health stage information in the online health community has some unique challenges and characteristics. First, though some patients share their disease history, as shown in Figure 3.1, such information is not provided or is simply missing for many others. For instance, over 36% active users that registered within recent 2 years have not yet shared their disease history in the Breast Cancer Community [2]. Important information about patients' health stages can significantly facilitate comprehensive investigations about patients' health conditions and thus it is highly desirable to be able to infer or predict these patients' health stage information. Second, different subforums under specific topics are often correlated to specific disease stages. For example, in the online breast cancer forum, the patients who are active in the "Chemotherapy - Before, During, and After" subforum typically look for infor-

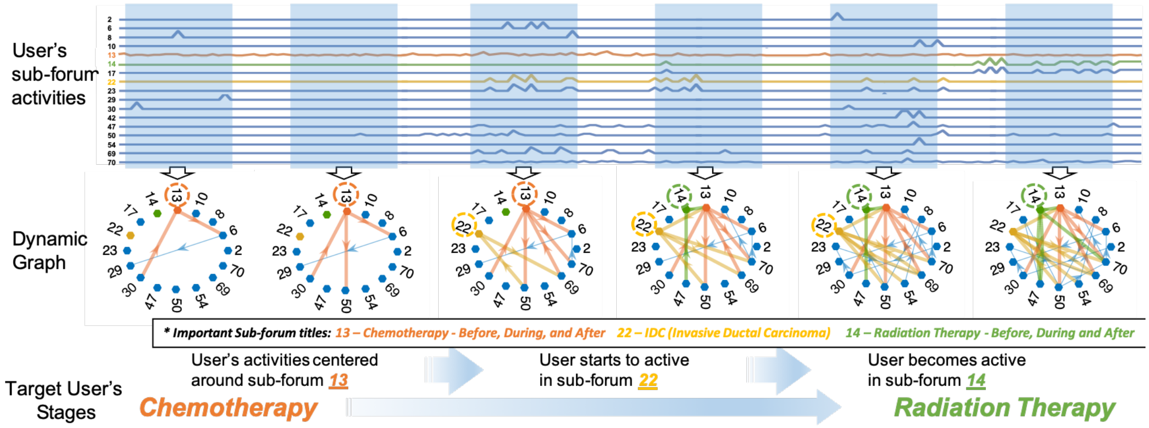

User's sub-forum activities
2
6
8
10
13
14
17
22
23
29
30
42
47
50
54
69
70
Dynamic Graph
13
10
8
6
2
70
69
54
50
47
30
29
23
22
17
14
* Important Sub-forum titles: 13 – Chemotherapy - Before, During, and After 22 – IDC (Invasive Ductal Carcinoma) 14 – Radiation Therapy - Before, During and After
Target User's Stages
User's activities centered around sub-forum 13
Chemotherapy
User starts to active in sub-forum 22
User becomes active in sub-forum 14
Radiation Therapy

Figure 3.2: An example of user forum activities and the corresponding health stage evolution. In the first two time windows, the user is mainly active in Subforum #13 while going through chemotherapy treatment. In the third time window, the user starts to be active in Subforum #22 at about the time when she undergoes IDC treatment. Finally, in the last three time windows, the user becomes active in Subforum #14 when she enters the "Radiation Therapy" health treatment stage.

mation related to their Chemotherapy treatment. Thus, users' activities within these subforums could serve as a strong indicator of an individual user's current health stage. Third, as the patients' health conditions progress over time, they often move from one set of subforums to others that are more related to their new health stages. Therefore, for each patient, these transitions among subforums can lead to an interconnected subforum activity network that evolves over time, which could be highly entangled with the progress of patient's health status or disease stage, as shown in Figure 3.2.

The ability to accurately infer users' missing health stage information is crucial, as this could enable health care organizations to better support patients by pinpointing the most valuable information for each at their particular health stage [66]. To infer the missing user health stage information, the correspondence between the users' forum activities and their health stage history needs to be accurately identified and modeled. Naturally, the networked and time-evolving forum activity data can be formulated as a dynamic sequence of user activity transition graphs that change over time. In addition, the target user health stage history can be formulated as a sequence prediction problem that needs to be inferred from their dynamic graph sequence. Thus, without loss of generality, a new generic task is presented here where the goal is to learn the mapping from a sequence of graph-structured data to a target sequence. In this paper, we limit our scope to the domain of online health forums and focus on health stage sequence prediction based on online health forums data.

However, capturing the high-level mapping between the evolution of the user activity networks and the changes in the corresponding user's health stage cannot be easily handled by existing techniques due to the following three challenges: **1) Difficulty in modeling the forum data, which is dynamic, networked, and multi-attributed.** A user's activities in the various subforums can change dynamically over time and these activity transitions naturally bridge different subforums. In addition, each subforum contains both unique and shared content, and identifying how this content is shared among subforums is important. **2) Difficulty in learning the association between a sequence of user activity networks and the corresponding sequence of health stages**. The sequence of user activity networks contains complicated graph-structured information that dynamically evolves over time. Developing end-to-end learning between such dynamic complex data and a specific sequence is highly difficult. **3) Lack of interpretability of the health stage sequence inference process.** The sequence of user activity networks has a two-level hierarchical structure, namely from the node (i.e., subforum) level to the network level, and from the network level to the health stage level. It is thus a major objective to incorporate this hierarchical structural information into the development of an interpretable health stage inference process.

In this paper, we formally define the generic learning problem of health stage sequence inference using online forum data and propose the first framework to address the aforementioned challenges effectively. The main contributions are as follows:

1. **Defining the novel problem of inferring user health stage information using online health forum data.** We define the health stage inference problem in online health forums and formulate the user activities as transition graphs that are capable of modeling user dynamic transitions between subforums and their complex relationships.

2. **Proposing a generic framework DynGraph2Seq for inferring target sequence from a sequence of graphs.** We propose a novel deep neural encoder-decoder framework for learning the mapping between complex dynamic graph sequence inputs and the target output sequence.

3. **Proposing dynamic graph regularization and a dynamic graph hierarchical attention mechanism for enhancing model effectiveness and interpretability.** We propose a dynamic graph regularization that enforces the smooth learning of consecutive graphs while preserving the heterogeneity across the graph sequence. In addition, we propose a new dynamic graph hierarchical attention mechanism that captures both the time-level and node-level attention, thus providing model transparency throughout the whole inference process.

4. **Conducting comprehensive experiments and case studies to validate the effectiveness and interpretability of the proposed model.** Experiments on online health forum dataset demonstrate that our proposed models outperform conventional sequence inference methods. In addition, our qualitative analyses and case studies provide interpretable insights into the learning results of the proposed model and its variations.

3.2 Related work

Our model draws inspiration from the research fields of online health community analysis, dynamic graph learning, hierarchical attention mechanisms, and neural encoder-decoder models.

3.2.1 Online Health Communities Analysis

A number of studies have focused on the analysis and utilization of online health communities data. Popular social media such as Twitter and Facebook are good for aggregate level pattern mining tasks such as discovering epidemic outbreaks [187] and other adverse events [152]. However, compared to specialized health forums such as the Breast Cancer Forum, their power is limited for discovering individual-level health stages and health network patterns due to the privacy issues involved and data scarcity. There have been several analyses of breast cancer forum data [155, 37, 181] and, more recently, machine learning models have been used for longitudinal analysis [182, 180] as well as for some binary classification tasks such as patient participation prediction [124] and cancer type classification [66]. However, we are the first to propose a general framework that can achieve health stage sequence inference using online forum data.

3.2.2 Dynamic Graph Representation Learning

As an emerging topic in the graph representation learning domain, dynamic graph learning has attracted a great deal of attention from researchers in recent years. Most of the current research falls into the dynamic graph embedding domain. Some of the proposed methods intuitively extend the idea from static graph embedding approaches by adding regularization [195, 185], while others propose specific models for capturing dynamic characteristics of the graph [147, 52, 193, 51]. There has also been some work on streaming graph learning [172, 39, 91], where network representations are learned dynamically as the network evolves. However, these graph embedding techniques typically focus on learning representations of the graphs, such as node embedding, but in many real-world applications the aim is to learn some high-level knowledge from the graph data, such as graph classification tasks [157, 158] and graph to sequence tasks [163, 165]. An end-to-end learning model is thus needed to learn the mapping between the whole sequence of graph data and the target output sequence, instead of merely focusing on learning node representations.

3.2.3 Hierarchical Attention Mechanism

The attention mechanism first proposed by [11] was used for machine translation tasks. Here, the attention mechanism was used to select reference words in the original language that matched specific words in the foreign language before translation. Luong et al. [89] extended the idea by proposing two simple and effective local attentional mechanisms. Feed-forward attention [112] was proposed as a simplified model of attention which is applicable to feed-forward neural networks. Later on, Hierarchical Attention Networks [167] was proposed to model the natural hierarchical structure of word-to-sentence and sentence-to-document level attention in document classification tasks. Raffel et al. [113] further improved the attention model with a monotonic attention constraint which assumes that the input sequence is processed in an explicitly left-to-right manner when generating the output sequence.

Besides being widely used for machine translation, the attention mechanism has also been introduced in the graph representation learning domain. Graph Attention Networks [148] introduced node-to-node attention mechanism for graph embedding, and many others followed and extended this idea [166, 4]. However, there is little to no work that focuses specifically on studying the unique hierarchical structure that is naturally present in dynamic graphs.

3.2.4 Neural Encoder-Decoder Models

The neural encoder-decoder models originally designed to solve neural machine translation problems [24, 143, 11, 89] have been widely extended to model the mapping of general object inputs to their corresponding sequences. A major focus has been on addressing the limitations of Seq2Seq when dealing with more complex objects, including Tree2Seq [38], Set2Seq [149], Recursive Neural Networks [138], and TreeStructured LSTM [145]. Recent advances in graph deep learning and graph convolutional networks have enabled researchers to utilize various graph deep learning models to handle challenges in the domains of machine translation and graph generation [16, 134, 83, 55]. Most recently, the graph2seq model [164, 79] was proposed as a general-purpose encoder-decoder model for graph-to-sequence learning, where no domain-specific information is needed. However, to the best of our knowledge, as yet there have been no reports of work that explores dynamic graph to sequence learning, where the natural sequential order contained in a dynamic graph and its sequences might be advantageous for neural encoder-decoder models.

3.3 Problem Formulation

3.3.1 User Forum Activities as a Dynamic Graph

The online forum data records the path of each user's transition from one subforum to another, as well as their activities within each subforum. In order to capture these complex transitions and model the relationships between subforums, we propose a novel method to formulate the raw user subforum activities into activity transition networks that preserve these characteristics.

An activity transition network is formulated naturally as follows. User activities are first partitioned into a series of time windows. We then begin by formulating

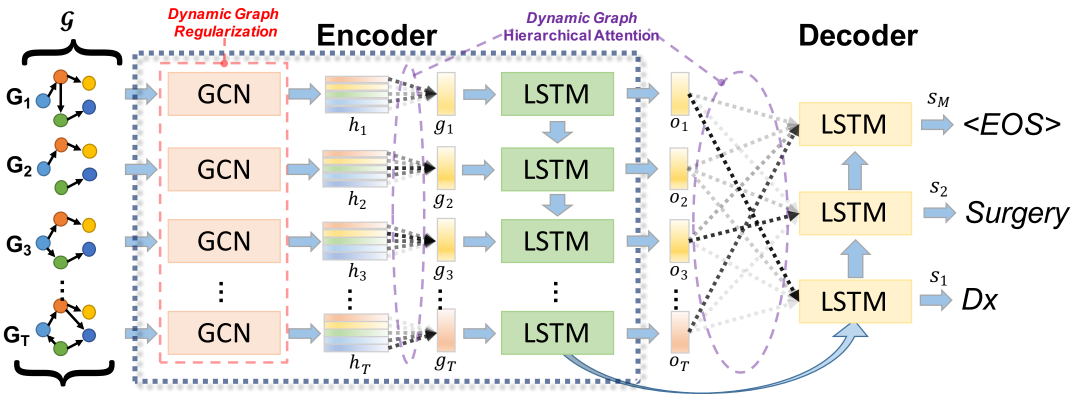


Figure 3.3: The proposed end-to-end dynamic graph-to-sequence learning (DynGraph2Seq) framework. It includes a novel dynamic graph encoder and a sequence decoder. The proposed framework not only generates sequence outputs by capturing complicated interactions of user's activities and dynamic characteristics of the evolving graphs over time, but also provides both time-level and subforum level interpretability of the correlations between a user's online forum activities and their current health stages through the proposed two-level attention mechanism.

a node for each subforum, with a transition from one forum to the other deemed to occur if the most active forum (based on visiting time or number of postings) switches from the former to the latter, creating a directed 'edge' between them. Each node (i.e., subforum) also records the user activity in the forum to build the activity transition network. For example in Figure 3.2, the subforum transition sequence is $\{30 \rightarrow 13 \rightarrow 6 \rightarrow 29\}$, where 30, 13, 6, and 29 are the IDs of the subforums visited. Thus, the transition edges for the first snapshot graph will be (30,13), (13,6), and (6,29). The graph in each time window records all the transitions in and previous to it.

Naturally, such time-ordered activity transition networks can be formally defined as dynamic graphs, also known as temporal networks in the network science literature [81], that capture the complex dynamic characteristics and time-evolving features of graphs, as defined in the following.

Definition 3. (dynamic graph). *A dynamic graph* $\mathcal{G} = \{G_1, G_2, \cdots, G_T\}$ *is an* *ordered sequence of* $t = 1, \cdots, T$ *separate graphs on the same set of* $|V| = N$ *nodes, with each snapshot graph* $G_t(V, E_t)$ *characterized by a weighted adjacency matrix* $A_t \in \mathbb{R}^{N \times N}$ *and a set of node features* $F_t \in \mathbb{R}^{N \times D}$ *for a given time window, where* D *represents the total number of node features.*

We can now formulate the activity transition networks as a dynamic graph, illustrated in Figure 3.2. Here, the dynamic graph contains a sequence of snapshot graphs $G_1, G_2, \cdots, G_6$ that characterize user activities in the online forum for a given time period, where G_t represents the snapshot graph $G_t(V, E_t)$ for simplicity. In this example, the time windows are shown as blue boxes. Each node $v \in V$ represents a subforum devoted to a specific topic and the edges E_t capture the user's movement between different subforums at a given time window. In addition, each node v contains a set of features $F_{t,v}$ that represents the topics covered by the specific subforum. By formulating user online forum activities as dynamic graphs, the mapping between the evolution of the user activity graphs and the changes of the corresponding user's health stages will be preserved.

3.3.2 Learning Sequence from Dynamic Graph

As we can see from Figure 3.2, there is a clear mapping between the evolution of the user activity dynamic graph and changes in the corresponding user's health stage. Motivated by this observation, we can formulate such problems as a general dynamic graph to sequence problem as follows:

Given a dynamic graph $\mathcal{G} = \{G_1, G_2, \cdots, G_T\}$ as input data, the goal is to predict the target sequence $S = \{s_1, s_2, \cdots, s_M\}$, where $s_m \in \mathbb{V}$ is the mth token of the output sequence in vocabulary $\mathbb{V}$; and T and M are the input graph sequence length and output sequence length, respectively. Formally, this problem is equivalent to learning a translation mapping from input dynamic graph $\mathcal{G}$ to a sequence S as $\{G_1, G_2, \cdots, G_T\} \rightarrow \{s_1, s_2, \cdots, s_M\}$.

The translation mapping problem between some source objects and target sequences has been widely studied, including both graph-to-sequence [164] and sequence-to-sequence [143, 24] formulations. However, dynamic-graph-to-sequence translation is more complex and poses several unique challenges, namely 1) Difficulty in comprehensively modeling the dynamic multi-attributed network-structured data, as both complex relationships and dynamic evolving characteristics need to be captured; 2) The temporal dependency of snapshot graphs in the dynamic graph need to be modeled and constrained by the learning model; and 3) The learned translation mapping is often obscure and hard to explain or verify. This is because the original low-level representation (i.e. the node level at a specific time) is aggregated into the high-level representation (i.e. the dynamic graph as a whole), making it much more difficult to backtrack and explain the correspondence.

3.4 Dynamic Graph-To-Sequence Model

3.4.1 The DynGraph2Seq framework

In this section, we introduce our dynamic graph-to-sequence framework that includes a novel dynamic graph encoder and a sequence decoder, as shown in Figure 3.3. To the best of our knowledge, this is the first end-to-end dynamic graph-to-sequence learning framework. Our DynGraph2Seq framework not only generates sequence outputs by capturing complicated interactions between a user's activities and the dynamic characteristics of the evolving graphs over time, but also provides both time-level and subforum level interpretability of the correlations between a user's online forum activities and that user's current health stages through our two-level attention mechanisms.

In order to capture the complex relationships represented in the graph input and the dynamic changes represented by the whole sequence of the dynamic graph, we propose a dynamic graph encoder that consists of three main components as follows: the first component contains a sequence of graph convolutional networks that learns the node level embeddings h_t for each graph snapshot G_t; the learned node level embeddings are then aggregated into a graph level embedding g_t by an aggregation function; finally, a sequence encoder is used to take the learned graph level embedding sequence $g = \{g_1, g_2, \cdots, g_T\}$ and generate a sequence of patient health stages that capture the entire dynamic graph characteristics. In addition, we propose a novel **dynamic graph regularization** for sparse feature selection of the graph convolutional networks that enforces smooth feature selection for consecutive snapshot graphs locally, while at the same time preserving the heterogeneity of the features selected across the entire dynamic graph globally.

Since our dynamic graph encoder is capable of learning the representation of the entire dynamic graph as a single global vector, we will be able to use a conventional sequence decoder as the decoder for our framework to generate the desired target sequence. Moreover, we propose a novel dynamic graph hierarchical attention mechanism that incorporates both **node-to-graph attention** and **graph-to-sequence attention** in order to promote better interpretability between graph sequences and output sequences and provide more effective aggregation function from node embeddings to graph embeddings. A detailed introduction to the proposed encoder and decoder will be described in the next two subsections.

3.4.2 Dynamic Graph Encoder

The base model of our graph convolutional network for each snapshot graph is inspired by graph2seq [164], which was originally proposed for addressing static graph-to-sequence learning problems. The Graph2Seq model employs an inductive node embedding algorithm that generates bi-directional node embeddings by aggregating information from a node local forward and backward neighborhood within k hops for a static graph. We extend this idea for dynamic graphs by applying such graph convolution on each snapshot graph within dynamic graph inputs. Specifically, suppose the total number of hops is k, then the hidden representation of n-th node in the snapshot graph G_t after applying the first graph convolutional layer will be computed as follows:

$$h^{\vdash}_{t,n} = mean(\{\sigma(W_t^{\vdash(1)} F_{t,u} + b_t^{\vdash(1)}), u \in \mathcal{N}_{\vdash}(v)\}) \tag{3.1}$$

$$h^{\dashv}_{t,n} = mean(\{\sigma(W_t^{\dashv(1)} F_{t,u} + b_t^{\dashv(1)}), u \in \mathcal{N}_{\dashv}(v)\}) \tag{3.2}$$

$$h^{(1)}_{t,n} = concat[h^{\vdash}_{t,n}, h^{\dashv}_{t,n}] \tag{3.3}$$

where $\mathcal{N}_{\vdash}(v)$ represents the set of forward neighbor nodes of node v, whereas $\mathcal{N}_{\dashv}(v)$ represents the set of backward neighbor nodes; $W_t^{\dashv(1)}, b_t^{\dashv(1)}$ and $W_t^{\vdash(1)}, b_t^{\vdash(1)}$ are learnable parameters for the first convolution layer. $F_{t,u}$ is the feature vector of node u in a snapshot graph at time step t; $\sigma(\cdot)$ represents the activation function of the network (e.g. ReLU); the $mean(\cdot)$ function takes the element-wise mean of the set of vectors in the equation; and $concat[vec1, vec2]$ concatenates the two row vectors into a single row vector.

Likewise, for hop k, the hidden representation of the n-th node in the snapshot graph G_t can be computed via the hidden representations computed from layer $k-1$, as follows:

$$h^{\vdash}_{t,n} = mean(\{\sigma(W_t^{\vdash(k)} h_{t,u}^{(k-1)} + b_t^{\vdash(k)}), u \in \mathcal{N}_{\vdash}(v)\}) \tag{3.4}$$

$$h^{\dashv}_{t,n} = mean(\{\sigma(W_t^{\dashv(k)} h_{t,u}^{(k-1)} + b_t^{\dashv(k)}), u \in \mathcal{N}_{\dashv}(v)\}) \tag{3.5}$$

$$h^{(k)}_{t,n} = concat[h^{\vdash}_{t,n}, h^{\dashv}_{t,n}] \tag{3.6}$$

Finally, after applying k layers of convolutions, the final hidden representation of the n-th node in the snapshot graph G_t will be output as $h_{t,n} = h_{t,n}^{(k)}$.

In order to capture the high-level representation of graphs for end-to-end graph learning, aggregating node level embeddings to graph level embedding that conveys the entire graph information is essential. To achieve this, we adopt the max pooling operation proposed by [164, 157] as the base aggregation function, which feeds the node embeddings $h_{t,n}$ to a fully-connected layer and then applies the max pooling method element-wise for each snapshot graph G_t to yield a sequence of graph-level representations g_t.

To model the graph dynamic changes and long-term dependencies throughout the M steps, we utilize Long Short Term Memory (LSTM) networks [59] as a graph embedding sequence encoder to learn the entire dynamic graph-level embedding. The computation of the LSTM network at time step t is defined as:

$$f_t = \sigma(W_f \cdot [o_{t-1}, g_t] + b_f) \tag{3.7}$$

$$in_t = \sigma(W_{in} \cdot [o_{t-1}, g_t] + b_{in}) \tag{3.8}$$

$$\widetilde{C}_t = tanh(W_C \cdot [o_{t-1}, g_t] + b_C) \tag{3.9}$$

$$C_t = f * C_{t-1} + in_t * \widetilde{C}_t \tag{3.10}$$

$$out_t = \sigma(W_{out} \cdot [o_{t-1}, g_t] + b_{out}) \tag{3.11}$$

$$o_t = out_t * tanh(C_t) \tag{3.12}$$

where o_t is the output of the LSTM network at time step t, C_t is the new cell state for the next time step computation, and the initial cell state for the encoder is set to all-zeros.

In the above encoder formulation, each graph convolutional network needs to learn a set of parameters for each snapshot graph in order to capture their unique characteristics. However, this will lead to several problems for the entire model during training: 1) the node embeddings learned from adjacent snapshot graphs G_t, G_{t+1} may yield inconsistent node embeddings even when the graph characteristics are similar, since there is no constraint on the parameter set; and 2) the resulting model tends to suffer from severe over-fitting issue since too many parameters need to be learned, especially when the total number of time steps T is large for a given dynamic graph.

Dynamic Graph Regularization for Sparse Feature Selection

To cope with the aforementioned challenges, we propose a novel temporal feature selection regularization that characterizes feature sparsity, local feature selection consistency, and global feature selection flexibility across the evolving graphs over time. Inspired by group sparsity $\ell_{2,1}$ regularization from group lasso [175] and overlapping group lasso [64], we propose the following dynamic graph regularization for the first layer of graph convolutional networks:

$$\mathcal{L}_{reg} = \beta \sum\nolimits_{t=1}^{T-w+1} \|\hat{W}^{(1)}_{[t:t+w]}\|_{2,1} \tag{3.13}$$

where w is the sliding window size; and $\hat{W}^{(1)}_{[t:t+w]}$ is the concatenated weight matrix from the weight parameters of a group of consecutive graph convolutional networks between time step t and $t + w$. Each row i of the weight matrix represents the ith feature weights across the w time steps; β controls the relative strength of the regularization.

Dynamic graph typically enjoys temporally consistent characteristics, since user transition activity graphs such as the example shown in Figure 3.2 change smoothly over time. Thus, the model can achieve temporal local consistency feature selection by adding a sliding window to force the local model to select similar features, which retains the flexibility of the feature selection process while still evolving gradually with time.

The proposed regularization brings several advantages. First, it promotes the interpretability of the model in term of node attributes, enabling us to visualize important features at any given time step and providing useful insights into how the importance of features evolves through time. Second, it also serves as a good regularizer to restrict the large number of model parameter sets, thus preventing possible model over-fitting. Lastly, it enhances the generalization power of the model. These results and analysis will be discussed in detail in the subsequent experimental section.

3.4.3 Sequence Decoder with Dynamic Graph Hierarchical Attention

Once the dynamic graph encoder takes the sequence of snapshot graphs G_t and aggregates node embeddings to generate a sequence of graph-level embeddings that capture the entire dynamic graph's global characteristics, the LSTM layer will output the final hidden-state of encoder C_T to summarize all the graph-level embeddings. Then, in the sequence decoding phase, we utilize a conventional sequence decoder [90] and set the initial cell state of the decoder as C_T in order to decode the target sequence S.

However, there are two issues with this simple sequence decoder: 1) the effectiveness of the sequence decoder depends on the length of the dynamic graph sequence (the longer the graph sequence is, the less information the last hidden state of graph embeddings can provide); and 2) the predicted user's health stage sequence need to be interpretable based on the dynamic graph sequence at both the time-level (i.e., which graph snapshot G_t is related) and node-level (i.e., which subforums this corresponds to). For instance, as shown in Figure 3.2, our model must learn to pinpoint which snapshot graphs in the dynamic graph sequence are strongly correlated to the output user health stage predicted by the decoder. To take this one step deeper, the model should also be able to provide information on which of the important nodes (subforums) are in a given snapshot graph while the model is generating a specific

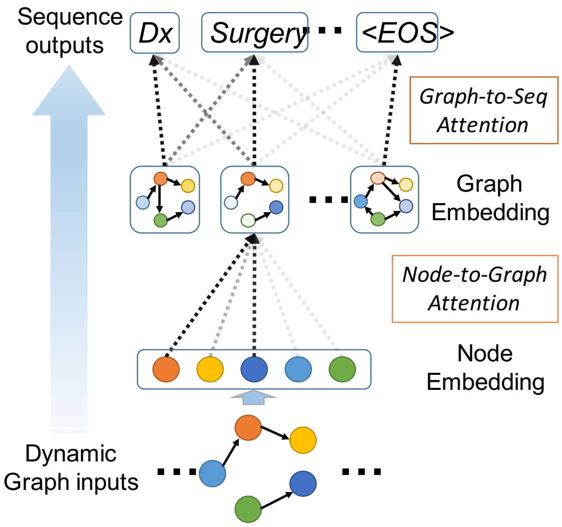


Figure 3.4: The proposed dynamic graph hierarchical attention mechanism: node-to-graph and graph-to-sequence attention. The node-to-graph attention aggregates the node level information (i.e. node embeddings) to formulate the graph-level embeddings, and the graph-to-sequence attention aims to find the mapping between each snapshot graph and each token in output sequence.

user's health stage.

To answer these questions pertaining to model interpretability, we need to develop a more effective way of handling information propagation and aggregation from low-level representations (i.e. node levels at a specific time) to high-level representations (i.e. the dynamic graph as a whole). Hence, we propose a novel dynamic graph hierarchical attention mechanism that includes **node-to-graph** and **graph-to-sequence** attention that is capable of enhancing the interpretability for node embedding aggregation and capture the hierarchical structure of user online forum activities over time more effectively.

Node-to-Graph Attention

Once the node embeddings of a graph have been computed, an average or max pooling operation [164, 157] is typically employed as the base aggregation function to obtain the graph-level embedding for the current graph. Although this works well in their individual settings, it does not work properly in our case since not all node embeddings contribute equally to the representation of the graph. For example, although a patient may view multiple subforums within a given time period, only a few important subforums will be correlated with the specific health stage of the patient. Therefore, it is vital to identify these important nodes (subforums) that contribute most to representing the embedding of the current graph. Inspired by [112], we adopt the feed-forward attention to aggregate the node embeddings and formulate the graph-level embeddings. Figure 3.4 shows an example of how the node-to-graph attention is computed for a snapshot graph G_t. For a given snapshot graph at step t, the *node-to-graph attention* is given as follows:

$$e_{t,n} = a(h_{t,n}) \tag{3.14}$$

$$\alpha_{t,n} = \frac{\exp\left(e_{t,n}\right)}{\sum_{k=1}^{N} \exp\left(e_{t,k}\right)} \tag{3.15}$$

$$g_t = \sum\nolimits_{n=1}^{N} \alpha_{t,n} h_{t,n} \tag{3.16}$$

where the function $a(\cdot)$ is a learnable function that depends on the node embeddings $h_{t,n}$; and g_t denotes the aggregated graph-level embedding for a snapshot graph at step t. In this formulation, the attention weights $\alpha_{t,n}$ explicitly model the importance of each node n when constructing the graph-level representation of g_t. Clearly, we can utilize the attention weight information for each node to pinpoint which nodes (subforums) are highly related to the current health stage. We will discuss the interpretability of our node-to-graph attention in detail in the experimental Section.

Graph-to-Sequence Attention

Once the graph-level embedding g_t has been obtained for each snapshot graph G_t, the whole sequence of graph embeddings $g = \{g_i\}_{i=1}^{T}$ is fed into the sequence decoder, which generates the global hidden embedding c that characterizes the entire sequence of dynamic graph information. Following the conventional encoder-decoder setup, c is set as the initial hidden state for the sequence decoder from which to generate the target sequence of the health stages.

Although the hidden vector c theoretically contains all the information needed for the decoder to generate the target sequence, the encoder's hidden representation o_t at each time step t also contains valuable information about the snapshot graph information at that time step during the sequence encoding. To reward graphs that are strongly correlated to the target sequence, we use the attention mechanism and introduce graph-to-sequence level attention to measure the importance of each snapshot graph with the target sequence. Specifically, as shown in Figure 3.4, the graph-to-sequence attention takes the sequence of hidden states for each graph $o = \{o_1, \cdots, o_T\}$ in the dynamic graph sequence as additional inputs to the decoder. This forces the decoder to consider both the current hidden state and the attention alignments between each word generated and for the whole sequence o. In addition, inspired by [113], we can also add a temporal monotonic constraint to enforce the attention alignment of snapshot graphs to ensure they are processed in an explicitly monotonic time order when generating the output sequence.

Therefore, suppose the decoder is at time step i and the hidden state of the previous step is represented as s_{i-1}, our graph-to-sequence attention is computed as follows:

$$e_{i,t} = a(s_{i-1}, o_t) \quad p_{i,t} = \sigma(e_{i,t}) \tag{3.17}$$

$$q_{i,t} = (1 - p_{i,t-1}) \cdot q_{i,t-1} + \alpha_{i-1,t} \tag{3.18}$$

$$\alpha_{i,t} = p_{i,t} q_{i,t} \quad r_i = \sum\nolimits_{t=1}^{T} \alpha_{i,t} o_t \tag{3.19}$$

where $q_{i,0} = 0$ and $p_{i,0} = 0$ for computing for the special case when $t = 1$; the context vector r_i is then used to compute the current hidden state in the decoder and generate a word in the target sequence.

3.5 Experiments

For this study, we evaluated the performance of our proposed model utilizing a real-world online health forum, namely the *breast cancer community*. We conducted comprehensive experiments with both quantitative evaluation and qualitative analyses of the learning results. All the experiments were conducted on a 64-bit machine with Intel(R) Xeon(R) W-2155 CPU 3.30GHz processor, 32GB memory and an NVIDIA TITAN Xp GPU.

3.5.1 Experimental Settings

Online Breast Cancer Community Dataset: The Breast Cancer Community [2] is one of the largest online forums designed for patients to share information related to breast cancer. So far, the forum has enrolled 215,671 registered members since the forum launch and the site contains a total of 81 subforums discussing 153,338 topics. The forum data collected for this study covers an 8 year period from the beginning of 2010 to the end of 2017. To create user subforum activity transition graph sequences, we defined user activities as being when they posted new topics or replied to existing topics and the time window was set as one month. After removing common words and stop words, we extracted the 100 top frequency keywords from the forum content to construct the feature vectors for the subforums. We randomly selected 70% of users who provided their health stage history for training, another 10% for validation, and the remaining 20% for testing. The predicted health stage sequences in the test data were validated against the real health stage history extracted from the corresponding users' signatures, as exemplified in Figure 3.1. The vocabulary of the health stages used in breast cancer consists of {‘Dx’[1], ‘Chemotherapy’, ‘Targeted’, ‘Hormonal’, ‘Radiation’, ‘Surgery’}.

Evaluation Metrics

We used BLEU scores [103] as the primary evaluation metric for determining the closeness of the model predicted health stage history and the ground truth. In addition, we also tested the model with ROUGE-1 score [85], which is commonly used for evaluating machine summarization and translation tasks.

Comparison Methods

NMT(seq2seq) The Neural Machine Translation model implemented by Luong et al. [90] is a widely used state-of-the-art sequence-to-sequence model for machine transition tasks. Since the NMT model can only handle simple sequence inputs, we simplified the input data by concatenating the transition sequences of user activity for each month together in time order. The subforum features are omitted in such formulations. We tested the model settings both with and without the attention mechanism.

Graph2seq The Graph2seq model [164] was recently proposed as a general-purpose encoder-decoder model for static graph to sequence learning. Since the model cannot handle dynamic graphs as input, we simplified the input by aggregating all the edges that appeared in the dynamic graph together into a single static graph. Again, we tested the model settings both with and without the attention mechanism.

Hyper-parameter Settings

For the models tested in this experiment, the Adam optimizer[69] was used with a learning rate of 0.001 and a batch size of 50 for model training; greedy search was

[1] Short for Oncotype DX test, an initial diagnosis that analyzes how a cancer is likely to behave and respond to treatment.

Table 3.1: Performance Evaluation for Health Stage Prediction. The scores were obtained from 20 individual runs and presented in a mean ± standard deviation (SD) format.

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE
NMT(seq2seq) (w/o att)	55.5±2.38	38.4±0.91	27.1±0.90	19.2±0.87	71.6±1.04
NMT(seq2seq) (w/ att)	57.8±1.86	40.4±1.21	29.0±1.28	20.1±1.06	72.9±0.86
Graph2Seq (w/o att)	57.5±1.72	41.5±0.94	29.8±0.72	20.3±0.85	75.8±1.20
Graph2Seq (w/ att)	58.2±2.19	41.1±1.38	30.1±0.83	21.0±0.51	76.2±0.96
DynGraph2Seq (w/o reg & att)	60.9±1.53	43.7±1.00	31.5±0.63	22.1±0.48	79.3±0.80
DynGraph2Seq (w/ reg)	61.5±2.42	45.1±1.86	32.3±1.31	23.1±1.05	78.5±0.86
DynGraph2Seq (w/ att)	62.3±1.46	44.7±1.29	32.0±0.94	22.5±1.13	80.8±0.36
DynGraph2Seq (w/ reg & att)	64.1±0.84	45.4±0.31	33.1±0.41	24.1±0.70	81.0±0.69

used for all the sequence decoders, selecting the highest scoring token at each stage. Hyper-parameters were searched and chosen based on the highest scores achieved on the validation set. For the graph encoders used in both the Graph2Seq and DynGraph2Seq models, the hop number k was set to 4. For the proposed dynamic graph regularization, the window size was set to 12 and β was set to 0.0003.

3.5.2 Performance

Table 3.1 shows the model performance of the baseline and proposed models. The scores were obtained from 20 individual runs and presented in a mean ± standard deviation (SD) format. In general, our proposed DynGraph2Seq framework significantly outperformed both the Seq2Seq and Graph2Seq baselines for the various model settings and evaluation metrics. The basic DynGraph2Seq framework with both the proposed dynamic graph regularization and the dynamic graph hierarchical attention achieved the best score on all the metrics, outperforming the baseline models by 10% - 25% on the BLEU scores and 6% - 13% on the ROUGE scores. The baseline Graph2Seq model also achieved good scores, but was not as competitive as our proposed model. This was largely because Graph2Seq model failed to capture the dynamic characteristics of user activity with only static graph inputs. The Seq2Seq model performed badly due to its inability to model the complex relationships between the subforums with simple sequence inputs.

Interestingly, although the full version of DynGraph2Seq (i.e. with the proposed regularization and hierarchical attentions) largely outperformed the baselines, the base model only achieved a marginal improvement compared to the Graph2Seq model. This was likely due to the fact that the aforementioned challenges prevented the base model from being fully effective for the learning task. These results further demonstrate that the proposed dynamic graph regularization and hierarchical attention are essential if the framework is to handle the learning tasks effectively.

3.5.3 Interpretablity Analysis

Interpretablity for dynamic graph hierarchical attention

Figure 3.5 shows an example of the learned dynamic graph hierarchical attention by DynGraph2Seq for test data. The left part of the figure shows the graph-to-sequence attention learned by the model, where each column is a grayscale heatmap representing the amount of attention being paid to each snapshot graph when the model predicted a specific health stage. The darker the color, the greater the attention being paid. We can see much attention was paid to the graphs around the months being labeled in the figure. The graphs for each labeled months are shown on the right. Interestingly, the graphs in the first two months attracted more attention from the model because those were the months when the patient first became active in the breast cancer online forum. The last two labeled snapshot graphs relate approximately to the time when the user engaged in extensive activities in a wide variety of subforums.

However, it is still hard to understand why these particular snapshot graphs were important and of interest to the model when predicting the user health stage sequence. To explore this issue, we went one step deeper by examining the node-to-graph level

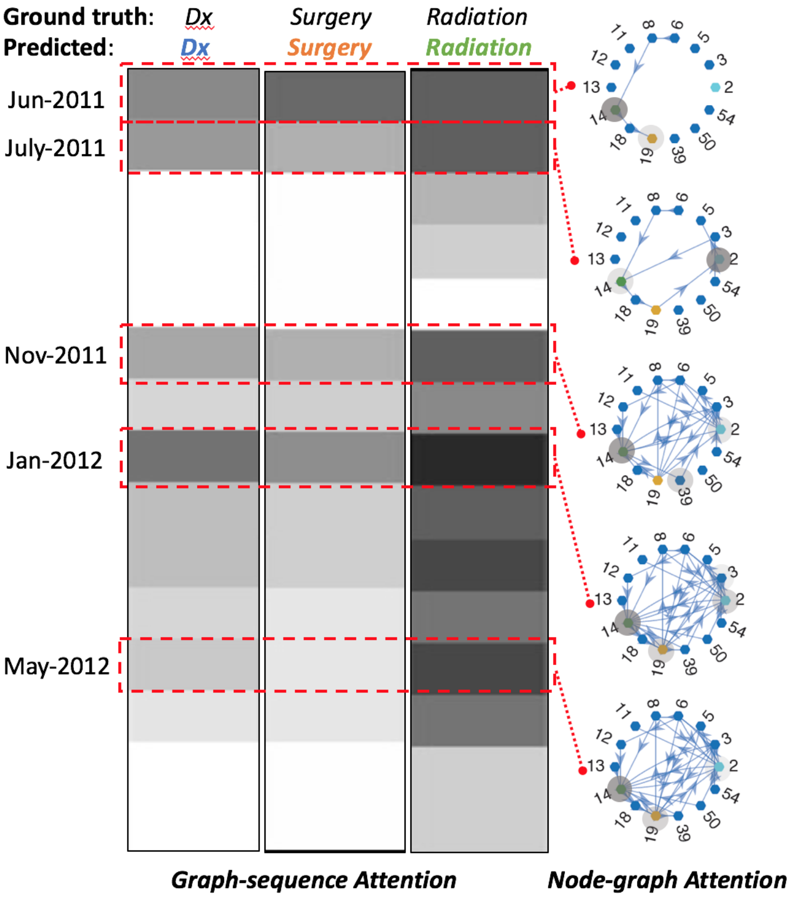


Figure 3.5: An example of learned dynamic graph hierarchical attention by DynGraph2Seq. The darker the color, the greater the attention being paid.

attention of these graphs. The red spots on the nodes shown on the right side of Figure 3.5 represent the amount of attention being paid to each node (i.e. subforum) when the model aggregated the node level information into the graph level representation. Again the darker the red spot, the greater the attention being paid. Now the attention becomes even more interesting and interpretable. For example, when constructing the representation of the May-2012 snapshot graph, Subforums #14, #19, and #2 received attention, with Subforum #14 being assigned the most attention. The title of Subforum #14 is actually "Radiation Therapy - Before, During and After", which is strongly correlated to the health stage 'Radiation'. This explains why that particular graph received more graph-to-sequence attention when the model predicted 'Radiation'. Likewise, we further discovered that Subforum #2, entitled "Not Diagnosed but Worried", has a strong correlation with 'Dx' and Subforum #19, entitled "DCIS (Ductal Carcinoma In Situ)", is a strong indicator for the model to predict 'Surgery'. These observed correspondences confirm that the proposed dynamic graph hierarchical attention mechanism greatly enhances the interpretability of the model.

Interpretablity for dynamic graph feature selection

Table 3.2 shows an example of the top subforum features (keywords) selected by the proposed dynamic graph regularization for the second half of the year 2017. The keywords in boldface were commonly selected during the year and exhibited high correspondences with patient health stage evolution. For instance, treatment-related keywords (e.g. 'diagnosed', 'treatment', 'therapy', and 'chemo') could be a strong indication of whether the patient was undergoing specific examinations or treatments. Moreover, the keywords containing temporal information, such as 'today', 'year', 'new', and 'newly', were also selected in many of the consecutive months. This is because these keywords could provide a temporal bridge to link dynamic graph sequences to the corresponding patient health stage sequences. Thus, the proposed

Table 3.2: Top 15 static subforum features (keywords) selected by dynamic graph regularization for the second half of the year 2017. The keywords in boldface are commonly selected during the year and have high correspondences with patient health stage evolution.

July	Aug	Sep	Oct	Nov	Dec
help	**bone**	bc	**today**	**help**	**bone**
sister	**treatment**	family	mets	**support**	**chemo**
treatment	**therapy**	sisters	scared	**new**	**help**
negative	scared	**year**	news	**chemo**	mets
today	mom	hair	**diagnosed**	results	please
bone	**diagnosed**	reconstruction	**support**	**bone**	**newly**
mom	family	care	negative	mets	bc
year	question	tumor	scan	please	survivor
please	radiation	positive	vs	ca	**diagnosed**
new	pain	lymph	results	pain	ladies
tumor	diagnosis	node	**therapy**	sisters	**support**
lymph	back	**treatment**	people	tamoxifen	anyone
share	looking	back	mom	**diagnosed**	**today**
lump	**new**	please	**new**	life	share
results	bc	research	**newly**	show	vs

dynamic graph regularization not only regularized the massive model parameters, but also brought significant benefits to enhance the model interpretability.

3.5.4 Health Stage Sequence Analysis

Correct Health Stage Sequence Predictions

Table 3.3 shows two examples of health stage sequences that DynGraph2Seq was able to infer correctly by capturing the dynamic evolution from the dynamic graph. In the first example, the patient underwent four surgeries, while in the second example the patient had two consecutive chemotherapy treatments. The baseline Graph2Seq failed miserably in terms of capturing such duplication due to the fact that a static transition network cannot preserve such information on the dynamic evolution of user forum activity.

Table 3.3: Correct Health Stage Sequence Predictions

Model	Health stage sequence
Ground Truth	***Dx Surgery Surgery Hormonal Surgery Surgery***
DynGraph2Seq	*Dx Surgery Surgery Hormonal Surgery Surgery*
Graph2Seq	*Dx Surgery Chemotherapy Hormonal*
Seq2Seq	*Dx Surgery Chemotherapy Hormonal Surgery*
Ground Truth	***Dx Chemotherapy Chemotherapy Surgery Radiation***
DynGraph2Seq	*Dx Chemotherapy Chemotherapy Surgery Radiation*
Graph2Seq	*Dx Surgery Chemotherapy Radiation*
Seq2Seq	*Dx Chemotherapy Surgery Surgery*

Table 3.4: Interesting Predictions on Complicated Sequences

Model	Health stage sequence
Ground Truth	***Dx Chemotherapy Surgery Hormonal Radiation Surgery***
DynGraph2Seq	*Dx Chemotherapy Surgery Radiation Hormonal Surgery*
Ground Truth	***Dx Surgery Radiation Dx Surgery Surgery Hormonal***
DynGraph2Seq	*Dx Surgery Radiation Dx Surgery Surgery Surgery Surgery Surgery*
Ground Truth	***Dx Dx Chemotherapy Hormonal Chemotherapy Hormonal Chemotherapy Chemotherapy Chemotherapy***
DynGraph2Seq	*Dx Surgery Chemotherapy Hormonal Dx Chemotherapy Chemotherapy Chemotherapy Chemotherapy Chemotherapy*

Interesting Incorrect Predictions

Exact sequence matching is extremely difficult, especially when inferring long and complicated sequences, such as the ones shown in Table 3.4. Surprisingly, even though DynGraph2Seq failed to predict exactly correct sequences, it generated meaningful health stage sequences that were very close to the ground truth. For instance, in the second case, DynGraph2Seq successfully predicted the first six stages, which involved two 'Dx' and three 'Surgery' stages. In the third case, which involved a total of 9 health stages, DynGraph2Seq was still able to predict reasonably close health stage sequences even for extremely long sequences. These cases further confirm that DynGraph2Seq did indeed succeed in learning meaningful patterns for this challenging dynamic graph sequence to sequence prediction task.

3.6 Conclusion

In this work, we formulated the task of health stage inference using online health forum data as a dynamic graph-to-sequence learning problem and propose a novel dynamic graph-to-sequence neural networks architecture (DynGraph2Seq) that can handle this new type of learning problem effectively. Our DynGraph2Seq model consists of a novel dynamic graph encoder and an interpretable sequence decoder to learn the mapping between a sequence of time-evolving user activity graphs and a sequence of target health stages. In addition, we developed a new dynamic graph regularization and dynamic graph hierarchical attention to facilitate the multi-level interpretability. Our comprehensive experiments and analyses for health stage prediction demonstrate both the effectiveness and the interpretability of the proposed models.

Chapter 4

Explanation-Guided Representation Learning on Geometric Data

4.1 Introduction

In recent years, representation learning on geometric data, such as image and graph-structured data, are experiencing rapid developments and achieving significant progress thanks to the rapid development of Deep Neural Networks (DNNs), including Convolutional Neural Networks (CNNs) and Graph Neural Networks (GNNs). However, DNNs typically offer very limited transparency, imposing significant challenges in observing and understanding when and why the models make successful/unsuccessful predictions [61].

While we are witnessing the fast growth of research in local explanation techniques in recent years, the majority of the focus is rather handling "how to generate the explanations", rather than understanding "whether the explanations are accurate/reasonable", "what if the explanations are inaccurate/unreasonable", and "how to adjust the model to generate more accurate/reasonable explanations" [13, 108, 168, 88, 129, 130, 62, 183, 151].

To explore and answer the above questions, this Chapter aims to explore a new line of research called 'Explanation-Guided Learning' (EGL) that intervenes the deep learning models' behavior through XAI techniques to jointly improve DNNs in terms of both their explainability and generalizability. Particularly, we propose to explore the EGL on geometric data, including image and graph-structured data, which are currently under-explored [61] in the research community due to the complexity and inherent challenges in geometric data explanation. Specifically, in Section 4.2, we introduce the proposed interactive and general explanation-guided learning framework GNES for graph neural networks to enable the "learning to explain" pipeline, such that more reasonable and steerable explanations could be provided. In Section 4.3, we describe the proposed two generic EGL frameworks, namely GRADIA and RES, for robust visual explanation-guided learning by developing novel explanation model objectives that can handle the noisy human annotation labels as the supervision signal with a theoretical justification of the benefit to model generalizability.

4.2 EGL on Graph-Structured Data

As Deep Neural Networks (DNNs) are widely deployed in sensitive application areas, recent years have seen an explosion of research in understanding how DNNs work under the hood (e.g., explainable AI, or XAI) [8, 5] and more importantly, how to improve DNNs using human knowledge [61]. In particular, Graph Neural Networks (GNNs) have been increasingly grabbed attention in several research fields, including computer vision [108, 43], natural language processing [7], medical domain [33], and beyond. Such trend is attributed to the practical implication of graphs data—many real-world data, such as social networks [40], chemical molecules [128], and financial

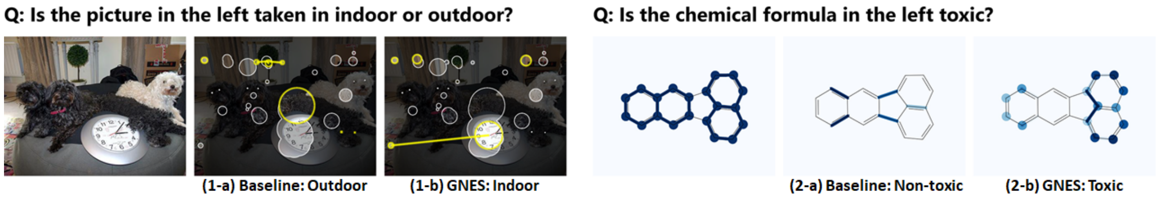

Q: Is the picture in the left taken in indoor or outdoor?
Q: Is the chemical formula in the left toxic?
(1-a) Baseline: Outdoor
(1-b) GNES: Indoor
(2-a) Baseline: Non-toxic
(2-b) GNES: Toxic

Figure 4.1: Cases for adjusting model explanation to improve Graph Neural Networks (GNNs). Scene Graph (left three): From the left, an input image, explanation before adjustment (1-a, inaccurate), and explanation after the adjustment (1-b, accurate). Note that the model explanation has been shifted from puppy eyes and back, rods, and an artificial tree to curtains, a clock, and a rug. Molecular formula (right three): From the left, an input formula, explanation before the adjustment (2-a, inaccurate), and explanation after the adjustment (2-b, accurate). Reactivity for this molecule is mostly affected by benzene ring sub-components in the overall molecular structure. 2-b highlights the main benzene rings of the molecule more effectively than 2-a.

data [96], are represented as graphs.

However, similar to other DNNs' architectures, GNNs also offer only limited transparency, imposing significant challenges in observing when GNNs make successful/unsuccessful predictions [61, 159]. This issue motivates a surge of recent research on GNN explanation techniques, including gradients-based methods, where the gradients are used to indicate the importance of different input features [13, 108]; perturbation-based methods, where an additional optimization step is typically used to find the important input that influences the model output the most with input perturbations [168, 88, 129]; response-based methods, where the output response signal is backpropagated as an importance score layer by layer until the input space [13, 108, 130]; surrogate-based methods, where the explanation obtained from an interpretable surrogate model that is trained to fit the original prediction is used to explain the original model [62, 183, 151]; and global explanation methods, where graph patterns are generated to maximize the predicted probability for a certain class and use such graph patterns to explain the class [173].

Despite the recent fast progress on GNN explanation techniques, the existing research body focuses on "how to generate GNN explanations" instead of "whether the

GNN explanations are inaccurate", "what if the explanations are inaccurate", and "how to adjust the model to generate more accurate explanations". Answering the above questions is highly beneficial to the model developers and the users of GNN explanation techniques, but are also extremely difficult due to several challenges: **1) Lack of an automatic learning framework for identifying and adjusting unreasonable explanations on GNNs.** Although there are plenty of existing works on GNN explanations, they are not able to ensure the correctness of explanations, not able to identify the incorrect explanations, nor able to adjust the unreasonable explanations. The technique that can enable this has not been well explored yet and is technically challenging due to the additional involvement of another backpropagation originated from explanation error. **2) Difficulty in aligning the node and edge explanations.** Existing GNN explanation works usually focus on either node and edge explanation while the interplay and consistency between the explanations of nodes and edges are extremely challenging to maintain and jointly adjusted. **3) Difficulty in jointly improving model performance and explainability with limited explanation-guided learning.** Due to the high cost for human annotation, it can be impractical to assume the full accessibility to the human explanation label during model training. Thus designing an effective framework that can best leverage a partially labeled dataset is on-demand yet challenging.

To address the above challenges, beyond merely generating GNN explanations, this paper focuses on a generic GNN explanation-guided learning framework for correcting the unreasonable explanations and learning how to explain GNNs correctly. Specifically, we first propose a unified explanation method for GNNs that can generate node and edge explanations with consistency regularization among them. The generality of the proposed method over existing node-explanation methods is rigorously demonstrated. Finally, we develop a learning objective that jointly optimizes model prediction and explanation with weak supervision from human explanation annotations.

Specifically, the main contributions of our study are as follows:

1. **Developing a generic framework for adaptively learning how to explain GNNs with weak explanation-guided learning.** We present a new learning objective for joint optimization among the model prediction loss, the explanation loss, and the graph regularization loss on regulating the model explanation. In addition, our framework can treat the explanation loss as an optional term and thus work effectively in scenarios where the human annotation on explanation is limited.

2. **Developing a unified graph-based explanation framework for calculating both node-level and edge-level explanation of GNNs.** We proposed a unified framework for both node-level and edge-level explanations that is suitable for explanation-guided learning and generalizable to the existing differentiable explanation methods.

3. **Proposing a model that can regularize both the node-level and edge-level explanations to form a better graph-level explanation.** We propose to apply novel explanation regularizations (i.e., explanation consistency and sparsity) onto the model-generated explanation to inject general graph principles and prior knowledge about the explanation that enhance the quality and consistency among the multiple levels of explanations.

4. **Conducting comprehensive experiments to validate the effectiveness of the proposed model.** Extensive experiments on five real-world datasets in two domains, chemical (molecular graphs) and vision (scene graphs), demonstrate that the proposed models improved the backbone GNN model both in terms of prediction power and explainability across different application domains. In addition, qualitative analyses, including case studies and user studies

of the model explanation, are provided to demonstrate the effectiveness of the proposed framework.

4.2.1 Related work

Our work draws inspiration from the research fields of graph neural network explanations that provide the model generated explanations, and explanation-guided learning on DNNs which enables the design of pipelines for the human-in-the-loop adjustment on the DNNs based on their explanations.

Graph Neural Networks Explanations

Most of the existing GNN explanation methods are instance-level methods, where the methods explain the models by identifying important input features for its prediction[174]. The first category is gradients-based methods, where the gradients are used to indicate the importance of different input features. Existing methods are SA [13], Guided BP [13], CAM [108], and GradCAM [108]. The second category is perturbation-based methods, where an additional optimization step is typically used to find the important input that influences the model output the most with input perturbations. Existing methods are GNNExplainer [168], PGExplainer [88], GraphMask [129]. The third category is the response-based method, where the output response signal is back-propagated as an importance score layer by layer until the input space. Existing methods in this category including LRP [13], Excitation BP [108] and GNN-LRP [130]. The last category is surrogate-based methods, where the explanation obtained from an interpretable surrogate model that is trained to fit the original prediction is used to explain the original model. The surrogate methods include GraphLime [62], RelEx [183], and PGM-Explainer [151]. Besides instance-level explanation methods, very recently, the global explanation of the GNN model has also been explored by XGNN [173]. Please see Yuan et. al. [174] for a survey of explainability in Graph

Neural Networks.

Even though there are plenty of existing explanation methods for GNNs, most of the methods above can not be applied to explanation-guided learning mechanism, as the goal is to apply supervision on the generated explanation such that the backbone GNN model itself can be fine-tuned accordingly to generate better explanations as well as keep or even improve the model performance. To enable this fine-tuning process over the explanation, the explanation itself needs to be differentiable to the backbone GNN model's parameters. In other words, only the explanation that is directly calculated from the computational pipeline (such as gradients-based and response-based methods) can be used to apply this additional explanation-guided learning to fine-tune the backbone GNN models explanation. The perturbation-based and surrogate-based methods all require additional optimization steps to obtain the explanation and thus are unable to be end-to-end trained with the explanation-guided learning on the backbone GNNs.

explanation-guided learning on DNNs

The potential of using *explanation*–methods devised for understanding which subparts in an instance are important for making a prediction–in improving DNNs has been studied in many domains across different applications. In fact, explanation-guided learning has been widely studied on image data by the computer vision community [87, 98, 111, 22, 105, 184, 32]. Linsey et al. [87] have demonstrated that the benefit of using stronger supervisory signals by teaching networks where to attend, which looks similar to the proposed approach. Moreover, Mitsuhara et al. [98] have proposed a post hoc fine-tuning strategy where an end-user is asked to manually edit the model's explanation to interactively adjust its output. Such edited explanations are then used as ground-truth explanations (from humans) to further fine-tune the model. In addition, several works in the Visual Question Answering (VQA) domain

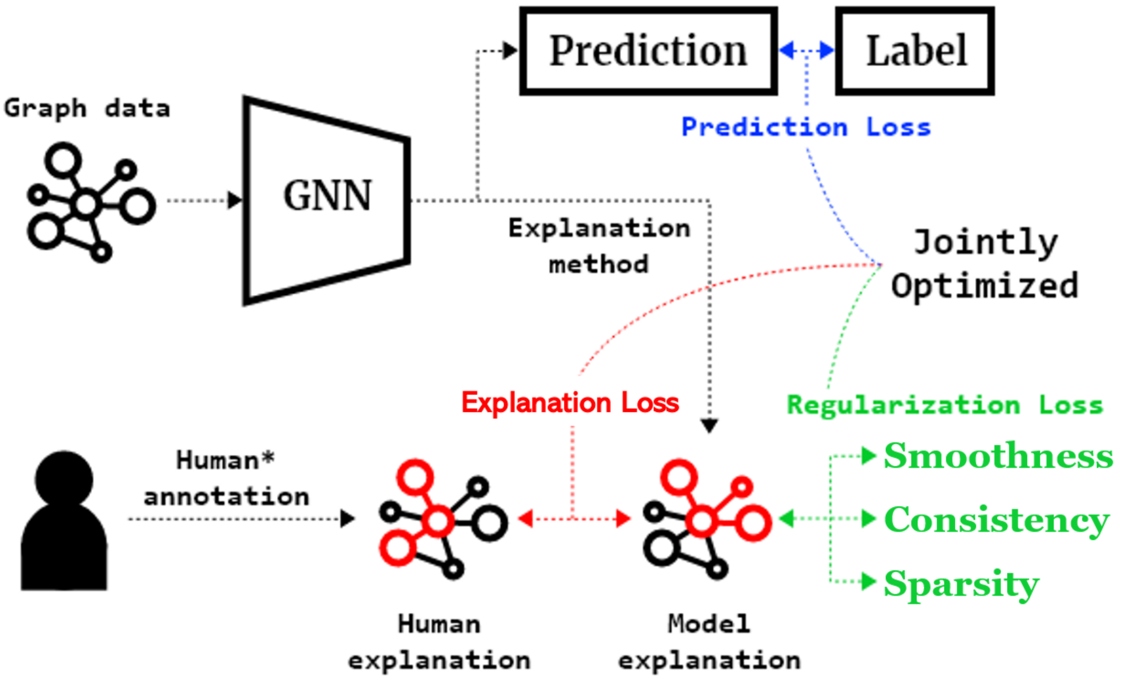


Figure 4.2: The proposed GNN explanation-guided learning (GNES) framework that jointly optimized the GNN models based on 1) a prediction loss, 2) an explanation loss on the human annotation and model explanation, and 3) a graph regularization loss to inject high-level principles of the graph-structured explanation. Notice that we only assume limited accessibility to the human annotation for only a small set of samples (10% in our experiments).

have proposed to use explanation-guided learning to obtain improved explanation on both the text data and the image data [111, 184, 105, 32]. Besides image data, the explanation-guided learning has also been studied on other data types, such as texts [65, 119], attributed data [150], and more. However, to our best knowledge, explanation-guided learning on graph-structured data with graph neural networks has not been explored before, and we are the first to propose a framework to handle this open research problem.

4.2.2 GNES Framework

In this section, we first introduce the proposed GNES framework that boosts the model explainability via explanation-guided learning and the novel explanation regularizations (i.e., explanation consistency and sparsity) that enhance the quality and consistency among the multiple levels of explanations. We then move on to introduce the proposed unified formulations for both node-level and edge-level explanation that are suitable for explanation-guided learning.

Problem formulation: Let $\mathcal{G} = (X, A)$ denote a attributed graph with N nodes be defined with its node attributes $X \in \mathbb{R}^{N \times d_{in}}$ and its adjacency matrix $A \in \mathbb{R}^{N \times N}$ (weighted or binary), where d_{in} denotes the dimension of input feature. Let y be the class label for graph $\mathcal{G}$, the general goal for a GNN model is to learn the mapping function f for each graph $\mathcal{G}$ to its corresponding label, $f : \mathcal{G} \rightarrow y$.

Following the definition of Graph Convolutional Networks (GCN) [70], a graph convolutional layer can be defined as:

$$F^{(l)} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}F^{(l-1)}W^{(l)}) \tag{4.1}$$

Where $F^{(l)}$ denotes the activations at layer l, and $F^{(0)} = X$; $\tilde{A} = A + I_N$ is the adjacency matrix with added self connections where $I_N \in \mathbb{R}^{N \times N}$ is the identity matrix; $\tilde{D}$ is the degree matrix of $\tilde{A}$, where $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$; The trainable weight matrix for layer l is denoted as $W^{(l)} \in \mathbb{R}^{d^{(l)} \times d^{(l+1)}}$; $\sigma(\cdot)$ is the element-wise nonlinear activation function.

In addition, to deal with variable size graphs in the dataset where the number of nodes can be different among graph samples, we adopt a similar design as in [108] to our backbone GNN model using several layers of graph convolutional layers followed by a global average pooling (GAP) layer over the graph nodes (e.g., atoms for the molecular graph and objects for the scene graph).

The Framework

The general goal for the GNES framework is to boost the model explainability via explanation-guided learning such that the model performance could also benefit from assigning more importance to the right features. Specifically, for graph data, the explanation-guided learning can be done in two main ways: 1) by applying some high-level graph-structured rules to the explanation, and 2) by adding human annotation samples as additional guidance. Thus, we present the learning objective of the GNES framework to be a joint optimization among the model prediction loss, the explanation loss, and graph regularizations on regulating the model explanation, as shown in Figure 4.2. Concretely, we propose the objective function as:

$$\min\ \mathcal{L}_{\text{Pred}} + \underbrace{\mathcal{L}_{\text{Att}}(\langle M, M'\rangle, \langle E, E'\rangle)}_{\text{explanation loss}} + \underbrace{\Omega(A, M, E)}_{\text{regularization}} \tag{4.2}$$

where $M \in \mathbb{R}^{N\times 1}$ and $E \in \mathbb{R}^{N\times N}$ denote the model generated node-level and edge-level explanations using a given explanation method. and M', E' are the corresponding ground-truth explanations marked by the human annotators. $\mathcal{L}_{\text{Pred}}$ is the typically prediction loss (such as the cross-entropy loss) on the training set. The proposed explanation loss $\mathcal{L}_{\text{Att}}$ measures the discrepancies between model and human explanations on both node-level and edge-level, as:

$$\mathcal{L}_{\text{Att}}(\langle M, M'\rangle, \langle E, E'\rangle) = \underbrace{\alpha_n \text{dist}(M, M')}_{\text{node-level loss}} + \underbrace{\alpha_e \text{dist}(E, E')}_{\text{edge-level loss}} \tag{4.3}$$

Where α_n and α_e are the scale factors for balancing node-level and edge-level loss; the function $\text{dist}(x, y)$ measures the mean element-wise distance between the inputs x and y, a common choose can be absolute difference or squared difference. In practice, we found that the absolute difference is more robust to the labeling noise from the annotator.

However, due to the high cost of human annotation on the explanations, obtaining the human explanations for the whole dataset can be prohibitive in practice. To deal with this issue, we propose to only apply the explanation loss to the samples that have the ground-truth labels for the human explanations, and apply the high-level graph rules to regulate the model explanation for each sample even if the human annotation is unavailable. Specifically, we propose a novel explanation consistency regularization term that regulates the node and edge explanation simultaneously so that the model is more likely to generate a globally consistent and smooth explanation over nodes and edges. Besides, we use sparsity regularization to regulate the model to only focus on a few important nodes and edges for the explanations. Thus, we propose the following graph regularizations to obtain more reasonable model explanations:

$$\Omega(A, M, E) = \underbrace{\beta\Omega_c(A, M, E)}_{\text{explanation consistency}} + \underbrace{\gamma\Omega_s(M, E)}_{\text{sparsity}} \tag{4.4}$$

Where β is the scaling factor for the explanation consistency between node and edge explanations, γ is the scaling factor for the sparsity constraints on both node and edge explanations. Concretely, each regularization and its desirable effects for regulating the graph explanation is described in more detail below:

Explanation consistency regularization. The node explanation and edge explanation are not independent, but rather highly correlated with each other. One natural assumption about the node explanation smoothness is that the adjacent nodes should share similar importance. However, this assumption can be too strong and sometimes lead to over-smoothing of the node explanation and tend to yield indistinguishable patterns for the explanation. In addition, it ignored the connection between the node and edge explanations, which can be a crucial factor for the explanation model to generate a global consistent explanation.

Here, we propose to take one step further regarding the smoothness assumption about the explanation by considering both node and edge explanations and making them more consistent with each other. Concretely, instead of treating all pairs of adjacent nodes equally important when enforcing the smoothness constraint, we propose to weight them by the corresponding edge importance such that the explanation consistency is better enforced on those nodes and edges that are deemed important. Mathematically, the explanation consistency can be measured by:

$$\Omega_c(A, M, E) = \frac{1}{2N^2} \sum_{i,j} E_{i,j} A_{i,j} \|M_i - M_j\|^2 \tag{4.5}$$

The above regularization can be interpreted as follows: given a pair of nodes i and j that is adjacent (i.e., $A_{i,j} = 1$), if the edge that connects the two nodes is important (i.e., $E_{i,j}$ is high), then the nodes it connects also tend to be consistent.

Sparsity regularization. As sparsity is a common practice for the model explanation, we apply the ℓ_1 norm to regulate both the node-level and the edge-level explanations, as:

$$\Omega_s(M, E) = \frac{1}{N}\|M\|_1 + \frac{1}{N^2}\|E\|_1 \tag{4.6}$$

Overall, the benefits of applying the proposed regularization terms are threefold. First, the regularization terms do not rely on the specific human labels on the explanation, which can be very limited and hard to acquire in practice. Thus they can be very crucial in the scenarios where the explanation labels are scarce. Second, since the explanation for the node and edge can be highly relevant, the proposed explanation consistency regularization can be critical for enforcing the model to generate more reasonable and consistent results that better align with the human explanation. Lastly, our overall framework is very flexible such that the regularization terms are not affected by changing the specification of the node and edge explanation formulation in Equation (4.7) and Equation (4.10), respectively, making the proposed framework easily applicable to give explanation and apply explanation-guided learning on any downstream applications with little to no overhead.

Node Explanation Formulation for explanation-guided learning

Although the node-level explanation is the most studied topic in the instance-based graph explanation domain, there are still several challenges to apply the node explanation-guided learning: First, most existing methods do not apply to the explanation-guided learning as the generated explanations are no longer differentiable to the backbone GNN model's parameters. Moreover, there is no unified formulation for the node-level explanation-guided learning.

To handle those challenges, we propose the first unified node explanation formulation for node-level explanation-guided learning. Concretely, we first identify that the gradient and the response/activation can be the major information that can produce the model-generated explanation that remains differentiable to the backbone GNN model's parameters so that the explanation-guided learning can be performed to affect the model during training. We then propose to integrate both aspects to form a general formulation for the node explanation. Mathematically, given the output y_c on class c, the explanation for node n at layer l can be computed as:

$$M_n^{(l)} = \|\mathrm{ReLU}(g(\frac{\partial y_c}{\partial F_n^{(l)}}) \cdot h(F_n^{(l)}))\| \tag{4.7}$$

Where $\frac{\partial y_c}{\partial F_n^{(l)}}$ represents the gradient of the features of node n at layer l given class c, and $F_n^{(l)}$ denotes the node activation at layer l, $g(\cdot)$ and $h(\cdot)$ are the functions that can be further defined to cover more complicated computation over the gradient as well as the activation, respectively.

The formulation above is a generic framework that covers as special cases major existing works where the gradient of the node features and the activation of the node are used to calculate the node explanation or the node importance, as shown in the following theorem.

Theorem 1 (Generality of Equation (4.7))**.** *The proposed generic node-level explanation formulation in Equation* (4.7) *covers a broad range of important existing works on node-level explanation as special cases with specification of* $h(\cdot)$ *and* $g(\cdot)$*, such as the gradient-based saliency maps (GRAD), GradCAM [131, 108], Layer-wise Relevance Propagation (LRP) [10, 13], and Excitation Backpropagation (EB) [177, 108].*

Proof. The specification for the function $g(\cdot)$ and function $h(\cdot)$ for each existing methods are listed in detail below:

Simple gradient-based saliency maps (GRAD): For simple GRAD, only the function $g(\cdot)$ is active, and it is simply the identity function, i.e. $g(\frac{\partial y_c}{\partial F_n^{(l)}}) = \frac{\partial y_c}{\partial F_n^{(l)}}$; the function $h(\cdot)$ will trivially return 1 (i.e. $h(F_n^{(l)}) = 1$) as the activation is not used in simple GRAD situation.

GradCAM: For the GradCAM [131, 108], since it uses both gradient information and node activation, both functions will be non-trivial. Specifically, the function $g(\cdot)$ can be defined as $g(\frac{\partial y_c}{\partial F_n^{(l)}}) = \frac{1}{N}\sum_{n=1}^{N}\frac{\partial y_c}{\partial F_n^{(l)}}$; and the function $h(\cdot)$ is the identity function (i.e. $h(F_n^{(l)}) = F_n^{(l)}$).

Layer-wise Relevance Propagation (LRP): For LRP [10, 13], gradient information is ignored and only the node activation is used. Concretely, the function $g(\cdot)$ will trivially return 1 (i.e. $g(\frac{\partial y_c}{\partial F_n^{(l)}}) = 1$); the function $h(F_n^{(l)}) = \frac{1}{d_l}\sum_{k=1}^{d_l}\hat{h}(F_{k,n}^{(l)})$ where $\hat{h}(F_{k,n}^{(l)})$ can be calculated via a relevance propagation as shown below.

For notational simplicity, we first decompose a graph convolutional operator into:

$$\begin{cases} \hat{F}_{k,n}^{(l)} = \sum_m V_{n,m}F_{k,m}^{(l)} \\ F_{k',n}^{(l+1)} = \sigma(\sum_{k'}\hat{F}_{k,n}^{(l)}W_{k,k'}^{(l)}), \end{cases} \tag{4.8}$$

where $V = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$ is the normalized graph Laplacian; the first equation is a local averaging of nodes, and the second equation is a fixed perceptron applied to each node (analogous to one-by-one convolutions in CNNs).

To capture both activatory and inhibitory parts of the forward pass, the $\alpha\beta-$rule is applied in RP, and the corresponding backward passes for these two functions can be defined as:

$$\begin{cases} \hat{h}(F^{(l)}_{k,n}) = \sum_m \frac{V_{n,m}F^{(l)}_{k,n}}{\sum_n V_{n,m}F^{(l)}_{k,m}} \hat{h}(\hat{F}^{(l)}_{k,m}) \\ \hat{h}(\hat{F}^{(l)}_{k,n}) = \sum_{k'}(\alpha \frac{\hat{F}^{(l)}_{k,n}W^{(l)+}_{k,k'}}{\sum_k \hat{F}^{(l)}_{k,n}W^{(l)+}_{k,k'}} + \beta \frac{\hat{F}^{(l)}_{k,n}W^{(l)-}_{k,k'}}{\sum_k \hat{F}^{(l)}_{k,n}W^{(l)-}_{k,k'}})\hat{h}(F^{(l+1)}_{k',n}), \end{cases} \tag{4.9}$$

where $W^{(l)+}_{k,k'} = max(0, W^{(l)}_{k,k'})$, and $W^{(l)-}_{k,k'} = min(0, W^{(l)}_{k,k'})$, and typically $\alpha + \beta = 1$ in order to uphold conservativity of relevance between layers.

Excitation Backpropagation (EB): For EB [177, 108], it follows the same setting as in LRP, except the parameter $\alpha = 1, \beta = 0$ in Equation (4.9), which only focus on the activatory or excitation part of the forward pass when calculating $h(F^{(l)}_n)$. □

Here we have demonstrated the broad coverage of the proposed node-level explanation formulation for enabling the unified node explanation-guided learning. Other existing gradient-based methods and response-based methods can also be easily derived and fitted into this framework by specifying the functions $g(\cdot)$ and $h(\cdot)$ respectively.

Edge Explanation Formulation for explanation-guided learning

Besides node-level explanation, the edge-level explanation can also be very crucial in many applications to highlight the important relationships between nodes. Unfortunately, most existing methods that focus on edge-level or subgraph-level explanations such as GNNExplainer [168], PGExplainer [88], and GraphMask [129] can not be used under the explanation-guided learning framework, as those explanations typically require additional objectives and optimization steps, making it not differentiable to the backbone model's parameters. Existing gradients-based methods and response-based methods typically focused only on node-level explanation, while little to no work has explored the edge-level explanation. Very recently, GNN-LRP [130] explored the higher-order edge-level explanation based on LRP. However, the multiple levels/orders of explanations on the edges are generally very hard to interpret and align with human annotations.

To enable edge-level explanation-guided learning, we propose the first unified edge-level explanation formulation following a similar path from node-level explanation. Concretely, using the chain rule, we identify that the gradient of the adjacency matrix, as well as the response/activation of the pairs of nodes that are associated with the edges can be the major information that can produce the model generated explanation that remains differentiable to the backbone GNN model's parameters. We then propose to integrate both aspects together to form a general formulation for the edge-level explanation. Concretely, given the output y_c on class c, the edge explanation between node n and node m at layer l can be computed as:

$$E_{n,m}^{(l)} = \|\mathrm{ReLU}(g(\frac{\partial y_c}{\partial F^{(l)}} \cdot \frac{\partial F^{(l)}}{\partial A_{n,m}}) \cdot h(F_n^{(l)}, F_m^{(l)}))\| \tag{4.10}$$

Where $\frac{\partial y_c}{\partial F^{(l)}} \cdot \frac{\partial F^{(l)}}{\partial A_{n,m}}$ represents the gradient of the edge that connects node n and node m at layer l given class c; $F_n^{(l)}$ and $F_m^{(l)}$ denote the activation of node n and node m at layer l, respectively; again $g(\cdot)$ and $h(\cdot)$ are the two functions that can be further defined to cover more complicated computation over the gradient as well as the activation, respectively.

Again, the formulation above should be able to generalize to most cases where the gradient of the edge and the activation of the pair of nodes are used to calculate the edge explanation. Although there is not yet any existing work that falls under this umbrella, we propose two possible specifications of the edge-level explanation from the above formulation as shown below.

Gradient-based: This can be seen as the extension from GRAD to edge-level explanation. Specifically, only the gradient information is used, as $g(\frac{\partial y_c}{\partial F^{(l)}} \cdot \frac{\partial F^{(l)}}{\partial A_{n,m}}) = \frac{\partial y_c}{\partial F^{(l)}} \cdot \frac{\partial F^{(l)}}{\partial A_{n,m}}$, and the node activation information is ignored, i.e. $h(F_n^{(l)}, F_m^{(l)}) = 1$.

Response-based: This can be seen as the extension from LRP to edge-level explanation. In this specification, the gradient information is ignored, i.e. $g(\cdot) = 1$, and the function $h(\cdot)$ is defined as:

$$h(F_n^{(l)}, F_m^{(l)}) = V_{n,m} \sum_{k=1}^{d_l} (\hat{h}(\hat{F}_{k,m}^{(l)}) + \hat{h}(\hat{F}_{k,n}^{(l)})) \tag{4.11}$$

where $\hat{h}(\hat{F}_{k,n}^{(l)})$ can be computed by Equation (4.8) and Equation (4.9).

4.2.3 Experiments

We test our GNES framework on two application domains, visual scene graphs and molecules. We first describe the detailed settings for the experiments and then present the quantitative studies on both model prediction as well as the explanation. In addition, we include several qualitative studies, including case studies and user studies, to make a qualitative assessment of how the proposed model has enhanced the explainability of the GNNs.

Experimental Settings

Molecular Graphs: We study three binary classification molecular datasets[1], BBBP, BACE, and task NR-ER from TOX21 [160], where the general goal for the classification task is identifying functional groups on organic molecules for biological molecular properties. Each dataset contains binary classifications of small organic molecules as determined by the experiment. The details of each dataset are listed below:

1. *BBBP*: The Blood-brain barrier penetration (BBBP) dataset comes from a recent study [95] on the modeling and prediction of barrier permeability. As a

[1] Available online at: http://moleculenet.ai/datasets-1

membrane separating circulating blood and brain extracellular fluid, the blood-brain barrier blocks most drugs, hormones, and neurotransmitters. Thus penetration of the barrier forms a long-standing issue in the development of drugs targeting the central nervous system. This dataset includes binary labels for 2053 compounds (graphs) on their permeability properties.

2. *BACE*: The BACE dataset provides quantitative (IC50) and qualitative (binary label) binding results for a set of inhibitors of human b-secretase 1 (BACE-1) [142]. This dataset contains a collection of 1522 compounds (graphs) with their 2D structures and binary labels.

3. *TOX21*: The "Toxicology in the 21st Century" (TOX21) initiative created a public database measuring the toxicity of compounds. The original dataset contains qualitative toxicity measurements for 8014 compounds (graphs) on 12 different tasks, here we selected the NR-ER task, which is concerned with the activation of the estrogen receptor [97].

Following the existing works on molecule classification [160], we split the dataset into train/validation/test with an 80/10/10 split ratio. In addition, we use the "scaffold" split algorithm for BBBP and BACE, where structurally similar molecules are partitioned in the same split. For TOX21, the random split is used.

Scene Graphs: We obtain the scene graphs from the Visual Genome dataset[2] [72]. The Visual Genome dataset consists of images and a corresponding scene graph where the nodes are objects in the scene and edges are relationships between objects. Objects and relationships are of many types and the data is collected from free-text responses obtained from crowd-sourced workers. Objects have an associated region of the image, defined by a bounding box. Following the previous work by [108], we construct two binary classification tasks: country vs. urban, and indoor vs. outdoor.

[2] Available online at: https://visualgenome.org/

The data samples for the two tasks are selected based on a set of pre-defined keywords which are used to query the Visual Genome data for matches in any attribute of an image. Specifically, the keywords used to define each class are listed below:

- *country*: countryside, farm, rural, cow, crops, sheep
- *urban*: urban, city, downtown, building
- *indoor*: indoor, room, office, bedroom, bathroom
- *outdoor*: outdoor, nature, outside

Notice that the keywords are non-comprehensive and the generated datasets are just for the purpose of studying the explanation on graphs. We balanced the sample size for each class by randomly selecting 1000 samples out of the image pools from the Keyword match. Again, we randomly split the dataset into train/validation/test with an 80/10/10 split ratio.

To convert the visual genome data to the graph input data, we treat each object as a unique node in the graph and the edge will be the corresponding relationship between a pair of objects. For the node feature for each object, we use a pre-trained InceptionV3 [144] network to extract the deep features from the image region defined by the bounding box associated with each object. The feature dimension for all visual genome nodes is of size $d = 2048$.

Evaluation Metrics: We evaluate the model in terms of performance as well as in terms of explainability. Specifically, for model performance assessment, we use accuracy (ACC) and Area Under the Curve (AUC) scores to measure the prediction power of the GNNs on the prediction tasks for sense graph datasets, and only AUC scores for molecular graph datasets as the sample size can be imbalanced. Besides, we leverage the human-labeled explanation on the test set to quantitatively assess the goodness of the model explanation. Specifically, for both node-level and edge-level explanations, we treat the human explanation as the gold standard, and compute the distance between human and model explanation via Mean Square Error (MSE) and Mean Absolute Error (MAE). To match with human annotation, both node-level and edge-level explanations are normalized in the range of $(0, 1]$ by dividing the corresponding max values.

Comparison Methods: Since there is no existing work on explanation-guided learning on GNNs and graph data, we demonstrate the effectiveness of our model in the following two aspects: First, we compare the explanation obtained by the proposed model with the explanation generated by the existing explanation methods on the backbone GNN as baselines to assess the improvement in terms of the model explainability. Concretely, we compare the explanation generated by GradCAM as the gradient propagation-based explanation, and EB as the relevance propagation-based explanation on a GNN with the same architecture as used in the proposed framework. Next, we conduct the ablation study of the proposed GNES framework to assess the effect of each proposed component. Specifically, we studied the following variations of GNES:

- $\textbf{GNES}_{+reg}^{-human}$: The variation where we ablate the human annotation and use graph regularization only to regulate the model explanation.
- $\textbf{GNES}_{-reg}^{+human}$ The variation where we ablate the regularization and only use the human annotation to supervise the model explanation.
- $\textbf{GNES}_{+reg}^{+human}$ The complete pipeline where we leverage both human annotation as well as graph regularization to supervise the model explanation.

Implementation details. Following the previous work on the explainability method on GNNs, we used a 3 layer GCN as our backbone GNN model. More specifically, the hidden dimension size for the three graph convolutional layers are of size 512, 256, and 128, respectively, followed by a global average pooling (GAP) layer,

Table 4.1: The performance and model generated explanation evaluation among the proposed models and the baselines on 3 molecular graph datasets. The results are obtained from 5 individual runs for every setting. The best results for each dataset are highlighted with boldface font and the second bests are underlined.

Dataset	Exp_Method	AUC	Node MSE	Node MAE	Edge MSE	Edge MAE
BBBP	EB	0.659 ± 0.011	0.572 ± 0.010	0.590 ± 0.009	0.050 ± 0.003	0.051 ± 0.002
	GradCAM	0.659 ± 0.011	0.460 ± 0.008	0.545 ± 0.004	0.042 ± 0.001	0.050 ± 0.001
	$\text{GNES}_{+reg}^{-human}$	0.662 ± 0.012	$\underline{0.375 \pm 0.018}$	$\underline{0.514 \pm 0.008}$	**0.029 ± 0.001**	**0.047 ± 0.001**
	$\text{GNES}_{-reg}^{+human}$	$\underline{0.665 \pm 0.009}$	0.449 ± 0.005	0.540 ± 0.006	0.041 ± 0.001	0.049 ± 0.001
	$\text{GNES}_{+reg}^{+human}$	**0.676 ± 0.007**	**0.358 ± 0.007**	**0.504 ± 0.007**	$\underline{0.032 \pm 0.001}$	$\underline{0.048 \pm 0.001}$
BACE	EB	0.703 ± 0.030	0.517 ± 0.008	0.548 ± 0.003	0.033 ± 0.001	**0.035 ± 0.000**
	GradCAM	0.703 ± 0.030	0.483 ± 0.006	0.544 ± 0.002	0.032 ± 0.000	$\underline{0.036 \pm 0.000}$
	$\text{GNES}_{+reg}^{-human}$	0.729 ± 0.009	0.427 ± 0.004	0.525 ± 0.002	0.026 ± 0.000	0.036 ± 0.000
	$\text{GNES}_{-reg}^{+human}$	$\underline{0.732 \pm 0.020}$	$\underline{0.421 \pm 0.004}$	$\underline{0.522 \pm 0.003}$	$\underline{0.024 \pm 0.001}$	**0.035 ± 0.000**
	$\text{GNES}_{+reg}^{+human}$	**0.733 ± 0.010**	**0.391 ± 0.005**	**0.519 ± 0.003**	**0.023 ± 0.001**	**0.035 ± 0.000**
TOX21	EB	0.788 ± 0.010	0.560 ± 0.028	0.622 ± 0.007	0.081 ± 0.006	0.091 ± 0.004
	GradCAM	0.788 ± 0.010	0.466 ± 0.018	0.566 ± 0.005	0.071 ± 0.002	0.084 ± 0.003
	$\text{GNES}_{+reg}^{-human}$	$\underline{0.789 \pm 0.020}$	0.460 ± 0.024	0.562 ± 0.004	$\underline{0.068 \pm 0.004}$	$\underline{0.081 \pm 0.001}$
	$\text{GNES}_{-reg}^{+human}$	0.789 ± 0.008	$\underline{0.393 \pm 0.009}$	$\underline{0.537 \pm 0.008}$	**0.065 ± 0.003**	0.083 ± 0.001
	$\text{GNES}_{+reg}^{+human}$	**0.794 ± 0.012**	**0.392 ± 0.008**	**0.523 ± 0.004**	**0.065 ± 0.002**	**0.079 ± 0.001**

and a softmax classifier. Models were trained for 100 epochs using the ADAM optimizer [69] with a learning rate of 0.001. The models were implemented in Keras with Tensorflow backend [26] and the newly proposed explanation loss and regularization loss were implemented by the custom loss function in Keras. We studied the node and edge explanation at the last GCN layer (i.e. $l = 3$). The node-level explanation for the GNES was specified following the GradCAM formulation, and the edge-level explanation is specified following the gradient-based formulation accordingly. The scale factors α_n and α_e for balancing node-level and edge-level loss in (4.3) were set to 1 by default; and the scale factors β and γ for the regularization in Equation (4.4) were grid researched via the AUC score on the validation set. Notice that for the human explanation annotation, we only used 10% of the human annotation for the training data for every dataset to simulate a more piratical situation where we only have partial human label data available. The samples in the test set are all labeled for evaluation purposes.

Performance

Table 4.1 shows the model performance and model generated explanation quality for the three molecular datasets. The results are obtained from 5 individual runs for every setting. The best results for each dataset are highlighted with boldface font and the second bests are underlined. For the models with human annotation (i.e., $\text{GNES}_{-reg}^{+human}$ and $\text{GNES}_{+reg}^{+human}$), we only assume 10% of the training sample has the explanation label for the node-level and edge-level explanations while all the remaining are treated as unlabeled samples. In general, our proposed GNES model variations outperformed the explanations from the backbone GNN model in terms of both prediction power as well as explainability on all 3 molecular datasets. More specifically, the ablation study of the model variations suggested that both the human annotation and graph regularization can have positive effects in different scenarios, and the full GNES model (i.e., $\text{GNES}_{+reg}^{+human}$) achieved the best performance, outperforming baseline GNN by 1% - 4% on AUC score. In addition, the full GNES model also significantly enhanced the explainability of the backbone GNNs by a great margin, both on node-level explanation (outperformed baselines by 20% - 37% and 6% - 16% on MSE and MAE, respectively) and on edge-level explanation (outperformed baselines by 9% - 36% and 1% - 13% on MSE and MAE, respectively). Those results demonstrated the effectiveness of the proposed framework not only on enhancing the model to pay correct explanation to the critical nodes and edges, but also consequently improved the model performance and prediction power on the prediction tasks.

Next, we examine the model performance and explanation quality on the two scene graph tasks. As shown in Table 4.2, all the setting are the same as in molecular graph tasks, except this time we also studied the accuracy (ACC) as the sample size for each class are balanced. We continue to see that the proposed GNES model achieved the best performance in terms of both ACC and AUC, and largely improved the GNN model's explainability on both node-level and edge-level explanations. Specifically, we

Table 4.2: The performance and model generated explanation evaluation among the proposed models and the baselines on 2 scene graph classification tasks. The results are obtained from 5 individual runs for every setting. The best results for each task are highlighted with boldface font and the second bests are underlined.

Dataset	Exp_Method	ACC	AUC	Node MSE	Node MAE	Edge MSE	Edge MAE
Indoor vs. Outdoor	EB	0.922 ± 0.009	**0.965 ± 0.001**	0.304 ± 0.002	0.361 ± 0.001	0.013 ± 0.000	0.016 ± 0.000
	GradCAM	0.922 ± 0.009	**0.965 ± 0.001**	0.280 ± 0.002	0.439 ± 0.006	0.010 ± 0.000	0.016 ± 0.000
	GNES$^{-human}_{+reg}$	0.927 ± 0.003	0.964 ± 0.002	0.274 ± 0.004	0.420 ± 0.007	0.010 ± 0.000	0.016 ± 0.000
	GNES$^{+human}_{-reg}$	0.925 ± 0.004	**0.965 ± 0.001**	0.270 ± 0.002	0.419 ± 0.005	0.010 ± 0.000	0.015 ± 0.000
	GNES$^{+human}_{+reg}$	**0.930 ± 0.005**	**0.965 ± 0.002**	**0.267 ± 0.003**	**0.406 ± 0.005**	**0.009 ± 0.000**	**0.014 ± 0.000**
Country vs. Urban	EB	0.991 ± 0.000	0.965 ± 0.003	0.271 ± 0.006	0.373 ± 0.008	0.015 ± 0.000	0.022 ± 0.000
	GradCAM	0.991 ± 0.000	0.965 ± 0.003	0.257 ± 0.006	0.433 ± 0.008	0.016 ± 0.000	0.023 ± 0.000
	GNES$^{-human}_{+reg}$	0.992 ± 0.000	0.965 ± 0.004	0.243 ± 0.001	0.414 ± 0.003	0.015 ± 0.000	0.022 ± 0.001
	GNES$^{+human}_{-reg}$	0.993 ± 0.000	0.969 ± 0.004	0.217 ± 0.008	0.347 ± 0.022	**0.014 ± 0.001**	**0.020 ± 0.001**
	GNES$^{+human}_{+reg}$	**0.994 ± 0.001**	**0.975 ± 0.005**	**0.212 ± 0.010**	**0.343 ± 0.020**	**0.014 ± 0.000**	**0.020 ± 0.001**

observed a 5%-22% improvement on node-level explanation, and a 7% - 30% improvement on edge-level explanation. All the above results have further demonstrated the general effectiveness of the proposed framework across different application domains.

Qualitative Analysis of the Explanation

Case Studies: Here we provide some case studies about the model explanation for both molecular graphs and scene graphs, as illustrated in Figure 4.3.

Molecular graphs: As shown in the bottom 3 rows of Figure 4.3, nodes and edges for molecular graphs were marked as important if they presented unique characteristics of significant reactivity or stability. For reactivity, special importance and annotations were provided if the atoms (nodes) and bonds (edges) were included in functional groups, highly polar bonds, and or groups with electron-donating and/or electron-withdrawing groups. Likewise, nodes and edges involved in resonance or conjugated systems that provide substantial electron delocalization (which are often attributes of highly stable compounds) were also indicated with high priority. Considering the examples from the TOX21 dataset at the last row of Figure 4.3, GNES is more accurate than Grad-CAM baseline in assessing the importance of the sulfonyl functional group and the corresponding resonance stabilization it experiences from the connected ring. Likewise, in the BACE example shown in the 4th row of Figure

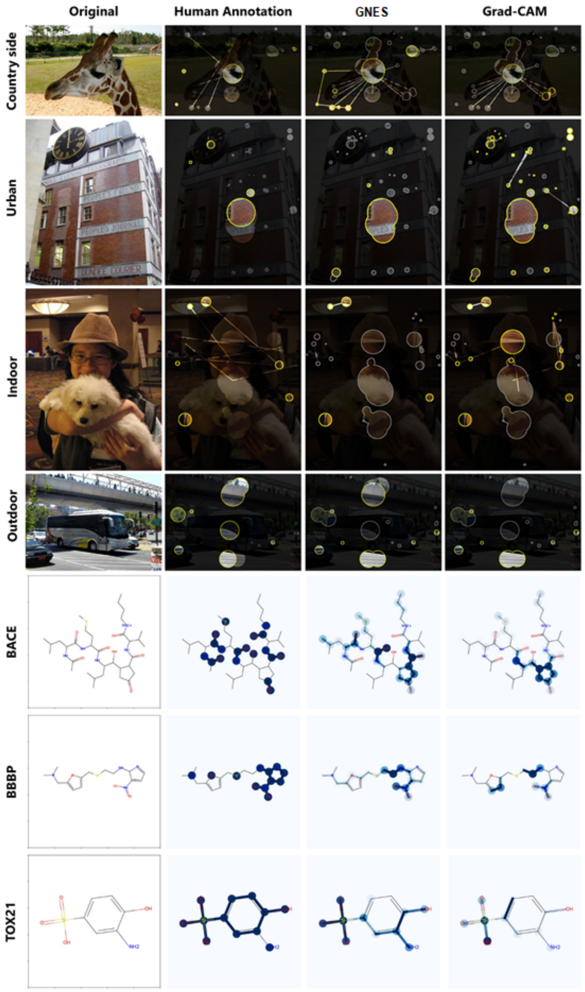


Figure 4.3: Selected explanation results for Scene graph dataset (top 4 rows) and molecule datasets (bottom 3 rows). For scene graph data, the size of the circle denotes the size of the bounding box of the object and the importance is marked by the lightness of the circle and the yellow boundaries. For molecule graphs, the importance is marked by the darkness of blue circles on nodes and blue lines on edges. Darker color means more importance is given.

Table 4.3: User study on scene graph datasets. The annotators were asked to give an overall evaluation specifically on the quality of the graph explanation (including both node-level and edge-level explanations). The final results were obtained by a joint work of 3 annotators.

Dataset	Exp_Method	# good	# bad	Positive rate
Indoor vs. Outdoor	EB	100	100	50.0%
	GradCAM	140	60	70.0%
	GNES	181	19	**90.5%**
Country vs. Urban	EB	96	94	50.5%
	GradCAM	140	50	73.7%
	GNES	165	25	**85.8%**

4.3, GNES has a better focus in highlighting functional groups and reducing priorities for irrelevant regions compared to the baselines models.

Scene graphs: As shown in the top 4 rows in Figure 4.3, for scene graph data, the size of the circle denotes the size of the bounding box of the object, and the importance of the nodes and the edges are marked by the lightness of the circles and lines, respectively. As can be seen, in general, the GNES model can more accurately focus on the importance of objects (nodes) and relationships (edges) than the Grad-CAM baselines. For example, as shown in the first row in Figure 4.3, the GNES explanation successfully found it is important to highlight not only the giraffe itself, but also the background (such as the fields) and the relationship between the giraffe and the fields. In contrast, the Grad-CAM baseline, however, only focused on the giraffe itself. Another example can be the indoor example at the 3rd row in Figure 4.3, and we can see that GNES gave more importance to the background objects and relationships, which are more accurate explanation and decisive factor for classifying this sample as the ”indoor” scene.

User Study Results on Scene Graphs: To further assess the quality and interpretability of the model generated explanation, we conducted a user study on scene graph datasets. The annotators were asked to give an overall evaluation of each of the model explanations, specifically focus on the quality and interpretability of the given explanation, including both node-level and edge-level explanation, as well as the consistency between the two as an overall explanation. The final results were obtained by a joint work of 3 annotators. Specifically, the process is as follows: the first annotator gives the initial assessment to all the samples considering only the graph explanation itself; then, after the first annotator finished labeling the dataset, the second annotator is asked to review the initial assessment and provide a list of samples he/she disagrees with the first annotator; finally, the third annotator will look into the list of samples where the first two have a disagreement on the label and make a final decision for those samples.

As shown in Table 4.3, we studied the quality for the two baseline explanations and our full framework (i.e., with both human annotation and graph regularization). As can be seen, our user study results further demonstrated that the proposed framework enhanced the GNN model's explainability by a huge margin. More specifically, our GNES model improved the quality of explanation on more than 40 (20%) samples in the test set of Indoor vs. outdoor datasets, and similarity turned more than 25 (13%) samples' explanation from bad to good quality. We argue that these results may have suggested that the GNES framework can have a big impact on the domains and applications, where the explainability of the machine learning model is crucial, and the data can be naturally presented in graphs/networks.

4.3 EGL on Image Data

As DNNs become available in a wide range of application areas, the study on explainability or explainable AI (XAI) is currently attracting considerable attention [5, 8, 54]. To open the "black box" of DNNs, many explainability techniques have been proposed that try to provide the "local explanation" of the DNNs prediction for a

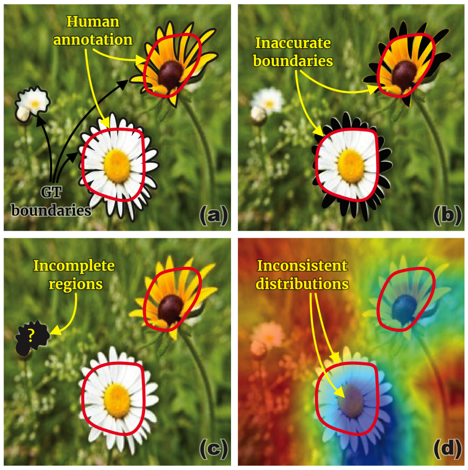


Figure 4.4: An example showing the challenges present in the human annotation labels: (a) human annotations are represented with red lines while ground-truth boundaries are shown with black lines. (b) Error caused by "inaccurate boundaries" are presented with black regions, (c) Error caused by "incomplete regions" are shown with a black region, and (d) the discrepancies between the "binary" human annotation and the "continuous" model-generated explanation maps. The explanation is queried based on predicting the scene as 'wild nature'.

specific instance [54], such as methods that provide the saliency maps for understanding which sub-parts (i.e., features) in an instance are most responsible for the model prediction [190, 131, 100, 10, 12, 99]. While we are witnessing the fast growth of research in local explanation techniques in recent years, the majority of focus is rather handling "how to generate the explanations", rather than understanding "whether the explanations are accurate/reasonable", "what if the explanations are inaccurate/unreasonable", and "how to adjust the model to generate more accurate/reasonable explanations".

Recently, techniques in *explanation-guided learning*, which support machine learning builders to improve their models by using supervision signals derived from explanation techniques, have started to show promising effects. The effects include improving both the generalizability and intrinsic interpretability of DNNs in many data types where the human annotation labels can be assigned accurately on each feature of the data. Such data type includes text data [65, 119] and attributed data [150].

However, the research on supervising explanations on image data—where the explanation is represented through saliency maps—is still under-explored [61]. In part, this is due to several inherent challenges in supervising visual explanations: **1) Inaccuracy of the human explanation annotation boundary.** It is difficult and costly for humans to make a perfectly accurate boundary which could lead the model to falsely assign positive explanation value to irrelevant features (i.e., pixels in image data). For example, as shown by the yellow arrows in Figure 4.4 (b), the coarsely drawn boundary falsely excluded a non-trivial region of the boundary of the wildflowers that could also be important to the prediction. **2) Incompleteness of the human explanation annotation region.** When labeling the explanation for image data, people usually tend to provide only a few regions as long as they are sufficient to convince people about the decision and do not bother to comprehensively find all the possible regions. Such incompleteness can mislead the model to wrongly penalize all the regions as long as they are not selected by annotators. Figure 4.4 (c) shows an example where the human annotation clearly missed one wildflower as shown in the black region. **3) Inconsistency of the data distribution between human annotation and model visual explanations.**The saliency maps generated by model explainers are continuous (e.g., Fig. 4.4 (d), heatmap) whereas human annotations are typically binary 'e.g., red circled areas annotated from humans in Fig. 4.4 (d) represent positive while the rest of areas are negative). Therefore, human-annotated explanations cannot be directly used to supervise the model and its explanations without significant efforts to fill the gap between the data domain and distributions.

To address the above challenges, beyond merely applying human annotation labels directly as the supervision signals to train the model, this work focuses on proposing a generic robust explanation-guided learning framework for learning to explain DNNs under the assumptions that the human annotation labels can be inaccurate in the boundary, incomplete in the region, as well as inconsistent with the distribution of the model explanation. Specifically, we propose a novel robust explanation loss that addresses all three aforementioned challenges present in the human annotation labels that can be noisy [27, 28]. In addition, we give a theoretical justification of the benefits of having the proposed explanation loss to the generalizability power of the backbone DNN model.

Specifically, the main contributions of our study are as follows:

1. **Proposing generic frameworks for learning to explain DNNs with explanation-guided learning.** We propose GRADIA and RES frameworks that enables visual explanation-guided learning on DNNs that is generalizable to the existing differentiable explanation methods.

2. **Developing a robust model objective that can handle the noisy human annotation labels as the supervision signal.** We propose a novel robust explanation loss that can handle the inaccurate boundary, incomplete region, as well as inconsistent distribution challenges in applying the noisy human annotation labels as the supervision signal.

3. **Providing a theoretical justification on the generalizability power of the proposed framework.** We formally derive a theorem that provides an upper bound for the generalization error of applying the proposed robust explanation loss when training the backbone DNN models.

4. **Conducting comprehensive quantitative and qualitative experimental analysis to validate the effectiveness of the proposed model.** Extensive experiments on two real-world image datasets, gender classification and scene recognition, demonstrate that the proposed framework improved the backbone DNNs both in terms of prediction power and explainability. In addition, qualitative analyses, including case studies and user studies of the model explanation, are provided to demonstrate the effectiveness of the proposed framework.

4.3.1 Related work

Our work draws inspiration from the research fields of local explainability techniques of DNNs that provide the model-generated explanation, and explanation-guided learning on DNNs which enables the design of pipelines for the human-in-the-loop adjustment on the DNNs based on their explanations to enhance both explainability and performance of DNN models.

Local Explainability Techniques of DNNs

As DNNs become widely deployed in a wide spectrum of application areas, recent years have seen an explosion of research in understanding how DNNs work under the hood (e.g., explainable AI, or XAI) [8, 54, 45, 171, 61]. Due to the "black box" nature of DNNs, most of the existing and well-received explainability methods focus on providing a "local explanation" that aims at explaining the prediction in understandable terms for humans for a specific instance or record [54]. One popular direction is to compute saliency maps as the local explanation, which provide the saliency values regarding which input features are most responsible for the prediction of the model [190, 131, 100, 10, 99]. For example, for image input, a saliency map is able to summarize where the model is "paying attention to" when performing a certain image recognition task. In this direction, one set of works incorporates network activations into their visualizations, such as Class Activation Mapping (CAM) [190] and Grad-CAM [131]. Another set of approaches takes a backward pass and assigns a relevance score for each layer backpropagating the effect of a decision up to the input level, existing works such as LRP [100, 10], and DTD [99] belong to this category. In addition, some model inspection methods such as VisualBackProp (VBP) [18] can also provide a local explanation similar to the LRP approaches. Besides the above techniques that are more specifically designed for interpreting image data, there are also several existing techniques that aim at providing more model-agnostic explanations on different types of data, such as LIME [115] and Anchors [116]. Please refer to the survey papers [8, 54] for a more comprehensive review of the existing works.

Explanation-Guided Learning on DNNs

The potential of using explanation–methods devised for understanding which subparts in an instance are important for making a prediction–in improving DNNs has been studied in many domains across different applications [46]. In particular, explanation-guided learning techniques have been widely explored on image data by the computer vision community [87, 98, 105, 184, 32]. Existing studies have shown the benefit of using stronger supervisory signals by teaching networks where to attend [87]. Following this line of study, several explanation-guided learning frameworks have been proposed. Mitsuhara et al. [98] proposed a post hoc fine-tuning strategy, where an end-user is asked to manually edit the model's explanation to interactively adjust its output. However, the proposed framework is only applicable to a specific type of DNN called Attention Branch Network [43]. In addition, several frameworks designed for the Visual Question Answering (VQA) domain have been proposed, where the goal is to obtain the improved explanation on both the text data and the image data [184, 105, 32].

Recently, several more generic frameworks have been proposed for explanation-guided learning on image data. One existing work proposed a conceptual framework HAICS [133], and the authors further implement it in an image classification application with human annotation in the form of scribble annotations as explanation-guided learning signals. Besides image data, the explanation-guided learning has also been studied on other data types, such as texts [65, 119, 25], attributed data [150], and more recently on graph-structured data [46]. However, most of the existing works typically assume the human labels are clean and accurate, while in practice they are

	Reasonable Attention	**Unreasonable Attention**
Accuracte Prediction	**RA:** Reasonable Accurate	**UA:** Unreasonable Accurate
Inaccurate Prediction	**RIA:** Reasonable Inaccurate	**UIA:** Unreasonable Inaccurate

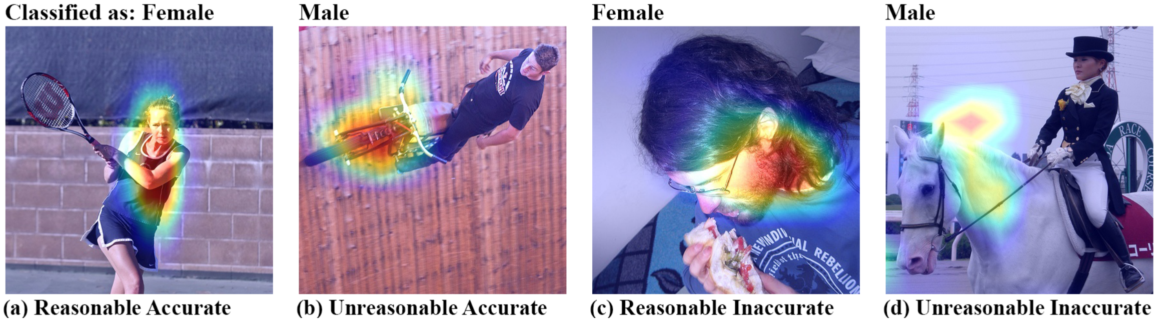

Classified as: Female
Male
Female
Male
(a) Reasonable Accurate
(b) Unreasonable Accurate
(c) Reasonable Inaccurate
(d) Unreasonable Inaccurate

Figure 4.5: **Reasonability Matrix** at the top with the four examples in a gender classification problem: (a) **Reasonable Accurate**: the attention given to an image is reasonable while prediction is also accurate, (b) **Unreasonable Accurate**: a substantial amount of attention is given to "contextual" features which make the attention unreasonable while the prediction is accurate, (c) **Reasonable Inaccurate**: despite the reasonable attention given to gender-intrinsic features, the prediction is not accurate, and (d) **Unreasonable Inaccurate**: the attention is unreasonable and the prediction not accurate.

prone to be inexact, inaccurate, and incomplete when directly used as the supervision signal for supervising the model explanation. To our best knowledge, we are the first to propose a robust explanation-guided learning framework that aims at handling this open research problem.

4.3.2 GRADIA Framework

We elaborate on our framework of IAAdevised for steering the way DNNs "think" based on human knowledge. Our framework has two novel components: (1) What to adjust: building of the Reasonability Matrixto systemically detect predictions made based on unreasonable/biased reasoning and adjust, and (2) How to adjust: applying GRADIAto leverage the adjusted attention maps in improving DNNs. Our framework is depicted in Fig. 4.6.

What to Adjust: Reasonability Matrix

The first stage in our framework aims at identifying the instances made based on biased reasoning. Based on the former work that demonstrates the benefit of considering model's reasoning via model explanation methods [58, 153], Reasonability Matrixelicits from human annotators regarding the *attention accuracy*: whether the model explanation given to an instance is *reasonable* for classifying the instance into a particular class. Specifically, we postulate that a human annotator can determine the whole, or some part of, attention given to an image is either *intrinsic attention*–the attention directly relevant for a classification–or *contextual attention*–the attention that shows "spurious correlation" between the object and a specific class (e.g., kitchenware and female, or a baseball bat and male). To help annotators to decide as to whether attention given to an instance is reasonable, we use the following two-step validation.

- **Q1. Intrinsic attention**: Is the attention given to an image presents sufficient details for a human annotator to classify the instance?
- **Q2. Contextual attention**: Using the attention given to an image, can a human annotator recognize any contextual objects?

We consider the given attention is reasonable when a human annotator answers positive for Q1 and negative for Q2. Combining the attention accuracy with conventional model accuracy, Reasonability Matrixleads to the four cases as follows:

- **RA. Reasonable Accurate**: The attention only focuses on intrinsic features without containing contextual features while the prediction result is also accurate (e.g., see Fig. 4.5 (a)).
- **UA. Unreasonable Accurate**: The prediction itself is accurate. But non-trivial amount of attention is given to contextual features, presumably due to contextual bias embedded in a training set (e.g., see Fig. 4.5 (b)).

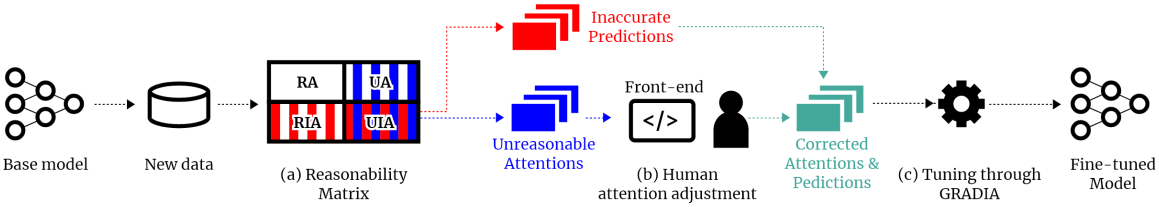


Figure 4.6: Overview of our methodological framework of interactive attention alignment. (a) Building Reasonability Matrix, (b) adjusting attention maps of inaccurate predictions & unreasonable instances, (c) fine-tune the model using GRADIA.

- **RIA. Reasonable Inaccurate**: While the attention is reasonable, the prediction is inaccurate. This might be caised by the lack of data points similar to this type in a training set (e.g., Fig. 4.5 (c) shows that attention is given to a man's beard but the model's prediction is inaccurate).

- **UIA. Unreasonable Inaccurate**: The attention is not reasonable and the prediction is also not accurate (e.g., see Fig. 4.5 (d)).

With the rise of the FaccT research, a broader ML community started to establish the consensus that heavily relying on a single performance metric, such as model accuracy, error score, or confusion matrix can be detrimental for a comprehensive capturing of a model's "crucial shortcomings" [102]. The capability of having the attention accuracy in structuring the Reasonability Matrixmeans that we can use the quality of attention as a new way to evaluate DNN's performance. On top of the widely used model prediction accuracy metric, our framework proposes the following metrics as additional ways to add more rigor in evaluating DNN:

- **P1. Reasonable Accurate Performance**: The metric that indicates the proportion of the "right answer based on the right reasoning" (i.e., $\frac{RA}{RA+UA+RIA+UIA}$) which is more rigorous than the commonly used model accuracy performance (i.e., $\frac{RA+UA}{RA+UA+RIA+UIA}$).

- **P2. Attention Accuracy Performance**: The metric explains the proportion of instances with accurate attention (i.e., $\frac{RA+RIA}{RA+UA+RIA+UIA}$). This metric can be a proxy that shows the quality of model attention.

How to adjust: GRADIA

Using the proposed Reasonability Matrix, our framework elicits adjusted attention from human annotators. In this section, we introduce how GRADIAuses the adjusted attention maps in fine-tuning DNNs. In addition to minimize the error in the original training set, our major goal is to also minimize the losses from the three terms UA, RIA, and UIA in the Reasonability Matrix, which directly leads to our objective:

$$\min \ \mathcal{L}_{\text{Train}} + \mathcal{L}_{\text{UA}} + \mathcal{L}_{\text{UIA}} + \mathcal{L}_{\text{RIA}} \tag{4.12}$$

where $\mathcal{L}_{\text{Train}}$ denotes the model prediction loss on the original training set; $\mathcal{L}_{\text{UA}}$, $\mathcal{L}_{\text{UIA}}$, and $\mathcal{L}_{\text{RIA}}$ measure the errors on Unreasonable Accurate (UA), Unreasonable Inaccurate (UIA), and Reasonable Inaccurate (RIA) samples in the Reasonability Matrixof validation set, respectively.

For each term in Equation (4.12), there are two types of losses, namely *prediction loss*, denoted by $\mathcal{L}^{(p)}$, and *attention loss*, denoted by $\mathcal{L}^{(a)}$. Considering that different term (from different quadrant in Reasonability Matrix) requires different focus and

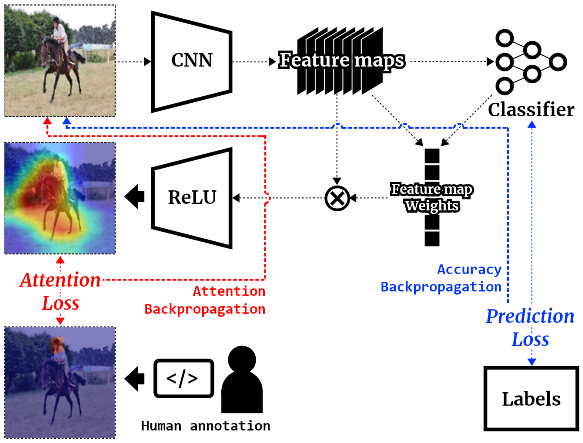


Figure 4.7: The computational pipeline of GRADIA.

balance between prediction and attention, we further introduce the balance factors for each term to give the model the flexibility to better weight between the attention and prediction loss in different cases. Specifically, Equation (4.12) can be expanded into the following one:

$$\min\ \mathcal{L}_{\text{Train}} + (\alpha\mathcal{L}^{(p)}_{\text{UA}} + (1-\alpha)\mathcal{L}^{(a)}_{\text{UA}}) + (\beta\mathcal{L}^{(p)}_{\text{UIA}} + (1-\beta)\mathcal{L}^{(a)}_{\text{UIA}}) + (\gamma\mathcal{L}^{(p)}_{\text{RIA}} + (1-\gamma)\mathcal{L}^{(a)}_{\text{RIA}}) \tag{4.13}$$

where the parameters α, β, and $\gamma \in [0,1]$ are the tunable factors for controlling the balance between the prediction loss and attention loss for UA, UIA, and RIA samples, respectively.

This way, the first term $\mathcal{L}_{\text{Train}}$ can also be expanded as a special case $\mathcal{L}_{\text{Train}} = \mathcal{L}^{(p)}_{\text{Train}}$ where the weight for $\mathcal{L}^{(p)}$ is set to 1 and the weight for $\mathcal{L}^{(a)}$ is set to 0, such that the attention map labels are not required. Finally, by further expanding the first term and rearranging the terms for prediction losses and attention losses, the final objective of GRADIAcan be written as:

$$\mathcal{L}_{\text{GRADIA}} = \underbrace{\mathcal{L}^{(p)}_{\text{Train}} + \alpha\mathcal{L}^{(p)}_{\text{UA}} + \beta\mathcal{L}^{(p)}_{\text{UIA}} + \gamma\mathcal{L}^{(p)}_{\text{RIA}}}_{\text{prediction loss}} + \underbrace{(1-\alpha)\mathcal{L}^{(a)}_{\text{UA}} + (1-\beta)\mathcal{L}^{(a)}_{\text{UIA}} + (1-\gamma)\mathcal{L}^{(a)}_{\text{RIA}}}_{\text{attention loss}} \tag{4.14}$$

where $\mathcal{L}^{(p)}$ can be calculated by applying the Cross-entropy loss on the corresponding samples of each terms; and $\mathcal{L}^{(a)}$ is the newly proposed attention loss that measure the attention quality of the samples.

Notice that since both the original training set and the new data samples are considered as a whole for the fine-tuning of the model (as shown by the prediction losses inside the 'prediction loss' bracket in Equation (4.14)), the above fine-tuning setup can naturally ensure the previously learned knowledge to be preserved and does not require freezing of the model parameters. Concretely, the prediction loss in Equation (4.14) consists of the prediction loss on the original training samples (i.e.

$\mathcal{L}^{(p)}_{\text{Train}}$) as well as new samples introduced in the fine-tuning stage (i.e. $\mathcal{L}^{(p)}_{\text{UA}}$, $\mathcal{L}^{(p)}_{\text{UIA}}$, and $\mathcal{L}^{(p)}_{\text{RIA}}$); while the attention loss consists of only the new samples introduced in the fine-tuning stage that has the human-adjusted attention labels available (i.e. $\mathcal{L}^{(a)}_{\text{UA}}$, $\mathcal{L}^{(a)}_{\text{UIA}}$, and $\mathcal{L}^{(a)}_{\text{RIA}}$).

Therefore, by introducing $\mathcal{L}^{(a)}$ into the fine-tuning step with GRADIA, the base DNN model can be jointly optimized both to generate higher quality attention maps and to make better and unbiased predictions on the original task. Our assumption is that this attention de-biasing process will also enhance the generalizability of the model to unseen data. As a result, GRADIAwill ultimately not only improve the model prediction accuracy, but also yield a more interpretable model.

To quantify the attention quality of the model, we propose a general attention loss for estimating the discrepancy between the model-generated attention maps and the human-annotated attention labels of the selected samples from the validation set. Concretely, the attention loss can be computed as the following:

$$\mathcal{L}^{(a)} = \text{dist}(M, M') \tag{4.15}$$

where M and M' are the model-generated attention maps and the ground truth attention maps provide by the human annotators on those samples that require attention adjustment; the function $\text{dist}(x, y)$ can be a common divergence metric such as absolute difference or square difference. In practice, we found that absolute difference is more robust to the labeling noise from the annotator, while square difference can be more sensitive and yield a high loss on the border areas of the labels that could not actually be related to the object.

To generate the model attention maps on images, several existing works have been proposed. Response-based methods such as CAM [190] and ABN [43] typically require substantial modification on the DNN architectures that either hurt the model's performance and extensibility or over-decouple the generation process of attention and prediction. For example, to handle the performance issue, ABN proposed to add another module called ‘attention branch’ onto the model architecture that is specialized for generating the attention maps. However, this incurs much more parameters and hence more samples and time to train the model. Moreover, over-decoupling the components for producing attention and prediction substantially decreases the reliability that the attention is indeed the explanation for the prediction. In contrast, gradient-based methods such as Grad-CAM [131] does not require changes of the base model and hence is applicable to a wide range of various DNN models. Moreover, it does not incur additional model parameters and hence can be more computationally cheap. Furthermore, its attention and prediction are tightly coupled and hence maintain a strong dependency and reliability between the prediction and its attention map.

Therefore, we propose to build our pipeline by extending Grad-CAM which uses the gradient of the feature maps with respect to the target class to generate the attention maps. Mathematically, suppose the penultimate layer produces K feature maps, $A^k \in \mathbb{R}^{u \times v}$ where u and k are the width and height of the image of each feature map. The attention maps $M_{\text{Grad-CAM}} \in \mathbb{R}^{u \times v}$ for target class c can be computed as:

$$M_{\text{Grad-CAM}} = \text{ReLU}(\frac{1}{uv} \sum_{k} \sum_{i} \sum_{j} \frac{\partial Y^c}{\partial A^k_{i,j}} \cdot A^k) \tag{4.16}$$

where Y^c denotes the output of the model for predicting class c, and $\sum_i \sum_j \partial Y^c / \partial A^k_{i,j}$ denotes the weight of the feature map k for class c as also illustrated by Figure 4.7. To ensure the generated and labeled attention maps are in the same scale, we further normalize $M_{\text{Grad-CAM}}$ to the values between 0 and 1, as:

$$M = \frac{M_{\text{Grad-CAM}} - \min(M_{\text{Grad-CAM}})}{\max(M_{\text{Grad-CAM}}) - \min(M_{\text{Grad-CAM}})} \tag{4.17}$$

where the function $min(\cdot)$ and $max(\cdot)$ return the element-wise min and max of the input, respectively.

4.3.3 RES Framework

In this section, we first introduce the proposed RES framework that enables explanation supervision on DNNs with both positive and negative explanation annotation labels. We then move on to propose a novel robust explanation loss that is designed to handle the inaccurate boundary, incomplete region, as well as inconsistent distribution challenges in applying the noisy human annotation labels as the supervision signal. Finally, we give the theoretical justification of the benefits of having the proposed explanation loss to the generalizability power of the backbone DNN model.

Problem formulation: Let $x \in \mathbb{R}^{C\times H\times W}$ be the input image data with C channels, H as height, and W as width. Let y be the class label for input x, the general goal for a DNN model is to learn the mapping function f for each input x to its corresponding label, $f : x \rightarrow y$.

The RES Framework

The general goal for the RES framework is to boost the model explainability via robust explanation supervision such that the model can robustly learn to assign more importance to the right input features even given noisy human explanation annotation labels, and consequently boost the task performance as well as the interpretability of the backbone DNN model. Here, we present the general learning objective of the RES framework to be a joint optimization of the model prediction loss and the robust explanation loss. Concretely, we propose the objective function as:

$$\min \sum_{i}^{N} \underbrace{\mathcal{L}_{\text{Pred}}(f(x^{(i)}), y^{(i)})}_{\text{prediction loss}} + \underbrace{\mathcal{L}_{\text{Exp}}(\langle M^{(i)}, F^{(i)}, C^{(i)}\rangle)}_{\text{robust explanation loss}} \tag{4.18}$$

where $M^{(i)} \in \mathbb{R}^{H \times W}$ denotes the model-generated explanations for ith sample using a given explanation method; $F^{(i)} \in \{0,1\}^{H \times W}$ and $C^{(i)} \in \{0,1\}^{H \times W}$ denote the corresponding binary labels for positive (i.e., $F^{(i)}_{j,k} = 1$ if the pixel at coordinate (j,k) of sample image i should be assigned with high importance, and 0 otherwise) and negative (i.e., $C^{(i)}_{j,k} = 1$ if the pixel at coordinate (j,k) of image i should be assigned with low importance value, and 0 otherwise) explanation marked by the human annotators. $\mathcal{L}_{\text{Pred}}(f(x^{(i)}), y^{(i)})$ is the typical prediction loss (such as the cross-entropy loss).

Robust Explanation Supervision for Noisy Explanation Annotation labels

To address the challenges presented in the noisy human annotation labels, we propose a robust explanation loss $\mathcal{L}_{\text{Exp}}$ that measures the discrepancies between model and human explanations regarding both the positive and negative explanation and taking into consideration the noisy nature of human annotation labels. Without loss of generality, let us assume $\tilde{M^{(i)}} = \tilde{F}^{(i)} - \tilde{C}^{(i)}$ in range $[-1,1]$ be the ground truth ideal explanation value for input image $x^{(i)}$, given the ideal positive explanation $\tilde{F}^{(i)} \in [0,1]$ and negative explanation $\tilde{C}^{(i)} \in [0,1]$; the binary human annotation as $F^{(i)}$ and $C^{(i)}$; and the model explanation as $M^{(i)} = g(f_\theta((x^{(i)}))$, where function $g(\cdot)$ specify the explanation method. We have $\mathbb{E}[\|M^{(i)} - (F^{(i)} - C^{(i)})\| - \|(F^{(i)} - C^{(i)}) - \tilde{M}^{(i)}\|] \leq \max\{0, \mathbb{E}[\|M^{(i)} - (F^{(i)} - C^{(i)})\|] - \mathbb{E}[\|(F^{(i)} - C^{(i)}) - \tilde{M}^{(i)}\|]\} \leq \mathbb{E}[\max\{0, \|M^{(i)} - (F^{(i)} - C^{(i)})\| - \|(F^{(i)} - C^{(i)}) - \tilde{M}^{(i)}\|\}] \leq \mathbb{E}[\|M^{(i)} - \tilde{M}^{(i)}\|]$ according triangle inequality. We define $\alpha = \mathbb{E}[\|(F^{(i)} - \tilde{F}^{(i)}) - (C^{(i)} - \tilde{C}^{(i)})\|]$. Therefore, to minimize $\|M^{(i)} - \tilde{M}^{(i)}\|$, we can have a tighter surrogate loss based on the annotated labels as follows:

$$\max\{0, \|M^{(i)} - (\tilde{F}^{(i)} - \tilde{C}^{(i)})\| - \alpha\}$$

Since the ground truth $\tilde{F}$ and $\tilde{C}$ are unknown, estimating α can be difficult. In practice, we can assume their distributions are positively correlated with the distribution of F and C, which can therefore be estimated by a slack variable α. To keep it simple and without loss of generality, in this work, we define α as a hyper-parameter of the framework assuming no additional knowledge about the ideal distribution.

Bridging the distribution between human labels and model explanation maps: To bridge the continuous model explanation $M^{(i)}$ with binary human labels C and F, we propose to split the above objective into two terms with bidirectional projections, as:

$$\min_{\theta,a} \sum\nolimits_{i}^{N} \max\{0, \|[\hat{M}^{(i)} - (F^{(i)} - C^{(i)})]\| - \alpha\} + d(M^{(i)}, h(F^{(i)}, C^{(i)})) \tag{4.19}$$

where $d(\cdot)$ is a distance function, $h(\cdot)$ is a mapping function that maps the binary masks $F^{(i)}$ and $C^{(i)}$ to continuous value in range $[0,1]$, and $\hat{M}^{(i)}$ is a binary projection of $M^{(i)}$ by a threshold a, as:

$$\hat{M}^{(i)} = \begin{cases} 1 & M^{(i)} \geq a \\ -1 & M^{(i)} < a \end{cases} \tag{4.20}$$

Basically, the above equation takes both the absolute difference (measured by the first term) and relative distance (measured by the second term) into consideration when comparing the continuous model explanation and the binary human explanation masks.

Mitigating the Inaccurate Boundary via Label Imputation: To realize the mapping function $h(\cdot)$ in Equation (4.20) which aims at projecting the binary human labels into continuous value domain, an intuitive way is to define $h(\cdot)$ as applying a $k \times k$ Gaussian kernel on the binary annotation labels F and C such that the pixels that close to the boundary of the manual label will also obtain slack values to boost the robustness and deal with the inexact and inaccurate boundary from human annotation.

However, a pre-defined kernel matrix might not be suitable for every data sample, and the discrepancy and inconsistency among annotators can also influence the accuracy of such a pre-defined estimation on handling the inaccurate boundary issue. Therefore, we further extend this idea and define a learnable imputation function $h_\phi(\cdot)$ with multiple learnable kernel transformations as the parameter set ϕ, such that the kernels' weights can be adjusted and learned to make better estimations of the ground truth explanation values and provide better mitigation to the inaccurate boundary problem. Specifically, the explanation loss with a learnable imputation function is as follows:

$$\min_{\theta,a,\phi} \sum\nolimits_{i}^{N} \max\{0, \|[\hat{M}^{(i)} - (F^{(i)} - C^{(i)})]\| - \alpha\} + d(M^{(i)}, h_\phi(F^{(i)}, C^{(i)})) \tag{4.21}$$

where ϕ is the parameter set of the imputation function $h_\phi(\cdot)$. The imputation function can be realized by applying multiple layers of convolution operations with learnable kernels over the raw annotation label F and C.

Handling the Incomplete Region by Selective Penalization

Finally, due to the incompleteness of human annotation labels, and to avoid falsely penalizing the model from assigning importance to the relevant features missed by the human labels, we propose to only selectively apply the explanation supervision signal onto the features with either positive or negative annotation labels. Concretely, we define the robust explanation loss $\mathcal{L}_{Exp}$ as follows:

$$\begin{aligned} \min_{\theta,a,\phi} \sum\nolimits_{i}^{N} & \max\{0, \|[\hat{M}^{(i)} - (F^{(i)} - C^{(i)})] \cdot \mathbf{1}(F^{(i)} - C^{(i)} \neq 0)\| - \alpha\} \\ & + d(M^{(i)} \cdot \mathbf{1}(F^{(i)} - C^{(i)} \neq 0), h_\phi(F^{(i)}, C^{(i)}) \cdot \mathbf{1}(F^{(i)} - C^{(i)} \neq 0)) \end{aligned} \tag{4.22}$$

where $\mathbf{1}(\cdot)$ is the indicator function, and $\cdot$ represents the elemental-wise multiplication operation. This formulation also gives the model a certain degree of flexibility on deciding the importance of unlabeled features based on data and downstream task, thus could yield a more generalizable and reasonable explanation that enhance both explainability as well as task performance of the model.

Optimization of Robust Explanation Loss

The indicator function for calculating $\hat{M}^{(i)}$ (as shown in Equation (4.20)) prevents us from directly optimizing our model objective with conventional gradient descent algorithms such as Adam [69]. Concretely, the optimization problem presented in Equation (4.22) involves optimizing both the adaptive threshold a and the model-generated explanation $M^{(i)} = g(f_\theta(x^{(i)}))$. Here, we propose to first find the optimal threshold a given model parameter θ, and then optimize θ with a conventional gradient descent algorithm by proposing a differentiable approximation to the indicator function.

First, to find the optimal a given θ, we need to solve the following objective:

$$\min_a \sum\nolimits_i^N \|[\hat{M}^{(i)} - (F^{(i)} - C^{(i)})] \cdot \mathbf{1}(F^{(i)} - C^{(i)} \neq 0)\| \tag{4.23}$$

Which is equivalent to the following by expanding $\hat{M}^{(i)}$:

$$\min_a \sum\nolimits_i^N \|[\mathbf{1}(M^{(i)} \geq a) - F^{(i)}] \cdot F^{(i)}\| + \|[\mathbf{1}(M^{(i)} < a) - C^{(i)}] \cdot C^{(i)}\| \tag{4.24}$$

If we treat each entry of $M^{(i)}$ as having two inequality constraints on a, we can efficiently solve the above formula in $O(m \log m)$ by our proposed algorithm by treating this optimization problem as finding a a that satisfies the maximum number of inequality constraints, where $m = max(|F|, |C|)$. The details of the proposed searching algorithm can be found in Appendix A.4.

To further enable gradient calculation of $M^{(i)}$ in Equation (4.22), we propose a surrogate loss using the hyperbolic tangent function $tanh(\cdot)$ to approximate the indicator function, as follows:

$$\min_{\theta,a,\phi} \sum\nolimits_{i}^{N} \max\{0, \|[tanh(\gamma(M^{(i)} - a)) - H^{(i)}] \cdot \mathbf{1}(H^{(i)} \neq 0)\| - \alpha\}$$
$$+ d(M^{(i)} \cdot \mathbf{1}(H^{(i)} \neq 0), h_{\phi}(F^{(i)}, C^{(i)}) \cdot \mathbf{1}(H^{(i)} \neq 0)) \tag{4.25}$$

where $H^{(i)} = F^{(i)} - C^{(i)}$; γ controls the slop of the hyperbolic tangent function. Moreover, when $\gamma \to \infty$, we can ensure such a approximation can be mathematically equivalent to the original indicator function in Equation (4.21) as shown in the following lemma.

Lemma 1. *Equation* (4.25) *is mathematically equivalent to Equation* (4.22) *when* $\gamma \to \infty$*.*

Proof. Please refer to Appendix A.2 for the proof. □

Theoretical Analysis of Generalizablity

We theoretically justify the generalizability power of the proposed explanation loss, as shown in Theorem 2 below.

We consider the regularized expected loss:

$$\mathcal{L}(f_\theta) = \mathbb{E}\left[\mathcal{L}_{\text{Pred}}(f_\theta(x), y) + \mathcal{L}_{\text{Exp}}(\nabla f_\theta(x))\right] \tag{4.26}$$

where f_θ is any learnable function with parameter $\theta \in \Theta$. In addition, denote the empirical loss as

$$\hat{\mathcal{L}}(f_\theta) = \frac{1}{N} \sum\nolimits_{i=1}^{N} \left(\mathcal{L}_{\text{Pred}}(f_\theta(x^{(i)}), y^{(i)}) + \mathcal{L}_{\text{Exp}}(\nabla f_\theta(x^{(i)}))\right) \tag{4.27}$$

where N denotes the training sample size. $\nabla f_\theta(x)$ denotes the gradient of f_θ on input

x, which can be used to generate any explanation. We omit the label (namely, $F^{(i)}$ and $C^{(i)}$) in $\mathcal{L}_{\text{Exp}}$ here for more compact notation. Also, we assume that $\mathcal{L}_{\text{Pred}}$ is L_1-Lipschitz and $\mathcal{L}_{\text{Exp}}$ is L_2-Lipschitz continuous w.r.t its first input, respectively.

Definition 4 (δ-minimizer)**.** A function $f_{\hat{\theta}}$ is said to be a δ-minimizer of $\mathcal{L}(\cdot)$ if

$$\mathcal{L}(f_{\hat{\theta}}) \leq \inf_{\theta \in \Theta} \mathcal{L}(f_\theta) + \delta \tag{4.28}$$

Assumption 1. Let f_{θ^*} be the solution to Eq. (4.26). There exists a neural network f_τ with $\tau \in \Theta$ such that

$$\|f_\tau - f_{\theta^*}\|^2 \coloneqq \mathbb{E}\left[|f_\tau - f_{\theta^*}|^2 + |\nabla f_\tau - \nabla f_{\theta^*}|^2\right] \leq C_1^2 \frac{\|\theta^*\|^2}{m^\gamma} \tag{4.29}$$

where C_1 is some constant, m is a constant related to the number of parameters in f, and γ is a constant order.

Assumption 2. Given any neural network f_θ from $\theta \in \Theta$ and i.i.d sample $\{x^{(i)}\}_{i=1}^N$. Given any $0 < \epsilon < 1$, we assume that

$$\sup_{\theta \in \Theta} |\mathcal{L}(f_\theta) - \hat{\mathcal{L}}(f_\theta)| \leq \frac{C_2(V, m, \epsilon)}{\sqrt{N}} \tag{4.30}$$

with probability at least $1 - \epsilon$. C_2 relies on set Θ, m and ϵ.

Such an inequality can be ontained using some statistical learning theories like Rademacher complexity.

Now we provide our generalization error bound as follow:

Theorem 2 (Generalizability of Equation (4.18))**.** *Let f_{θ^*} be the minimizer of $\mathcal{L}(\cdot)$, $f_{\hat{\theta}}$ be a δ-minimizer of $\hat{\mathcal{L}}$, then given $0 < \epsilon < 1$, with probability at least $1 - \epsilon$ over* *the choiec of $x^{(i)}$, we have*

$$0 \leq \mathcal{L}(f_{\hat{\theta}}) - \mathcal{L}(f_{\theta^*}) \leq (L_1 + L_2)\frac{C_1\|\theta^*\|}{m^{\gamma/2}} + \frac{2C_2(V, m, \epsilon)}{\sqrt{N}} + 2\delta \tag{4.31}$$

Proof. Please refer to Appendix A.1 for the formal proof. □

Our Theorem 2 provides an upper bound for the generalization error between the numerical optimal solution $\hat{\theta}$ and the theoretical optimal solution θ^*. The first term in the bound corresponds to the approximation error given in the first assumption, the second term corresponds to the quadrature error given in the second assumption, and the last term corresponds to the training error. To reduce the generalization error, we need to increase both the number of parameters and training samples. Meanwhile, the empirical loss is needed to be solved sufficiently well.

4.3.4 Experiments

We test our RES framework on two application domains, gender classification and scene recognition. We first describe the detailed settings for the experiments and then present the quantitative studies on both model prediction as well as the explanation. In addition, we include several qualitative studies, including case studies and user studies, to make a better qualitative assessment of how the proposed model has enhanced the explainability of the backbone DNN models.

Experimental Settings

Gender Classification Dataset: The gender classification[3] is one of the widely used tasks in the research of fairness in broader machine learning communities [186, 14, 58]. We constructed the dataset from the Microsoft COCO dataset[4] [86] by extracting

[3] We are aware that using a binary classification in gender does not reflect on the diverse viewpoint of gender in the real world, and we emphasize that the binary "gender classification" task here does not represent our viewpoint on gender.

[4] Available online at: https://cocodataset.org/

images that had the word “men” or “women” in their captions. We then filtered out instances that 1) contain both words, 2) include more than two people, or 3) humans appear in the figure is nearly not recognizable from human eyes. We collected a total of 1,600 images that satisfied our criterion and obtained the human annotation labels for all the image samples with our human annotation UI (please refer to Appendix A.3 for more details). For data splitting, we only randomly sampled 100 samples out of the 1,600 images as the training set to better simulate a more practical situation where we only have limited assess to the human explanation labels. The rest of the 1,500 data samples were then evenly split as the validation set and test set.

Scene Recognition Dataset: We obtained the scene images from the Places365 dataset[5] [191]. The original dataset contains more than 10 million images comprising 400+ unique scene categories. Following the macro-class defined by [191], we constructed a binary scene recognition task: nature vs. urban. The data samples for the two classes were randomly sampled from a set of pre-defined categories under macro-class “nature” and “urban”, respectively. Specifically, the categories we used to sample the data are listed below:

- *Nature*: mountain, pond, waterfall, field wild, forest broadleaf, rainforest
- *Urban*: house, bridge, campus, tower, street, driveway

Notice that the categories are non-comprehensive and the generated datasets are just for the purpose of studying the quality of model explanation. We balanced the sample size for each category and collected a total of 1,600 images. Again, we obtained the human annotation labels for all the samples with the human annotation UI, and split the data randomly with sample sizes of 100/750/750 for training, validation, and testing.

Evaluation Metrics: We evaluate the model in terms of task performance as well as in terms of explainability. For model performance, we use the conventional

[5] Available online at: http://places2.csail.mit.edu/index.html

prediction accuracy to measure the prediction power of the backbone DNN models as the datasets studied are well imbalanced. For explainability assessment, we leverage the human-labeled explanation on the test set to assess the quality of the model explanation. Specifically, we use the Intersection over Union (IoU) score [15], which is calculated by taking the bit-wise intersection and union operations between the ground truth explanation and the binarized model explanation to measure how well the two explanation masks overlap. In addition, since the IoU score only assesses the quality of positive explanation, we further compute the precision, recall, and F1-score as additional metrics which provide a more comprehensive evaluation of the model-generated explanation by considering the alignment of both positive and negative explanation.

Comparison methods: We compare the performance of the RES framework with the vanilla backbone model as the baseline as well as two existing explanation supervision methods, GRADIA [48] and HAICS [133]. For the proposed framework, we show two variations: RES-G and RES-L, with different implementations of the imputation function. Concretely, we studied the following methods:

- **Baseline**: The conventional DNN model that is trained with only the prediction loss.

- **GRADIA** [48]: The proposed framework that trains the DNN model with both the prediction loss as well as a conventional L1 loss that directly minimizes the distance between the continuous model explanation and the binary positive explanation labels.

- **HAICS** [133]: A framework that trains the DNN model with both the prediction loss as well as a conventional Binary Cross-Entropy (BCE) loss that directly minimizes the distance between the continuous model explanation and the combination of positive and negative binary explanation labels.

- **RES-G** [47]: The proposed RES framework with the imputation function $g(\cdot)$ as a fixed value Gaussian convolution filter.
- **RES-L** [47]: The proposed RES framework with the learnable imputation function $g_\phi(\cdot)$ via multiple layers of learnable kernels.

Implementation Details: For all the methods studied in this work, the backbone DNN model is based on the pre-trained ResNet50 architecture [56]. All models were trained for 50 epochs using the ADAM optimizer [69] with a learning rate of 0.0001. To make a fair comparison on explainability, the model explanations were all generated by the well-recognized explanation technique GradCAM [131], although other local explanation techniques can also be applied in our framework. The generated explanation maps are normalized in the range of $(0, 1]$ by dividing the maximum saliency value on each sample for model training as well as visualization. When calculating the explanation evaluation metrics, the explanation maps were further binarized by a fixed threshold of 0.5. The hyper-parameter α of the proposed RES framework was set to 0.001 for the gender classification task, and 0.01 for the scene recognition task, based on grid research via prediction accuracy on the validation set. The detailed implementation of the imputation layers for RES-L can be found in the Appendix A.5.

Performance

Table 4.4 shows the model performance and model-generated explanation quality for gender classification and scene recognition datasets. The results are obtained from 5 individual runs for every setting. The best results for each dataset are highlighted with boldface font and the second bests are underlined. In general, our proposed framework variations, i.e., RES-G and RES-L, outperformed all other comparison methods in terms of both prediction accuracy as well as explainability on both datasets. Specifically, regarding prediction power, the RES-G with a pre-defined Gaussian transforma-

Table 4.4: The performance and model-generated explanation evaluation among the proposed models and the comparison methods on both gender classification and scenes recognition tasks. The results are obtained from 5 individual runs for every setting. The best results for each task are highlighted with boldface font and the second bests are underlined.

Dataset	Model	Accuracy	IoU	Precision	Recall	F1
Gender Classification	Baseline	68.35 ± 1.00	13.68 ± 0.89	52.68 ± 0.61	56.34 ± 1.63	47.77 ± 1.14
	GRADIA	70.01 ± 1.47	16.66 ± 1.10	64.07 ± 2.07	51.84 ± 3.55	53.35 ± 3.08
	HAICS	69.29 ± 0.50	17.56 ± 0.79	60.06 ± 2.17	56.48 ± 2.13	54.90 ± 2.14
	RES-G	**71.33 ± 0.53**	22.97 ± 0.44	**76.47 ± 0.45**	63.90 ± 3.64	63.54 ± 2.29
	RES-L	70.39 ± 0.35	**23.60 ± 0.36**	76.32 ± 0.77	**65.75 ± 1.20**	**65.24 ± 0.74**
Scene Recognition	Baseline	93.42 ± 0.43	38.55 ± 0.22	**89.67 ± 0.07**	60.96 ± 0.56	68.47 ± 0.46
	GRADIA	95.03 ± 0.35	39.60 ± 1.13	87.98 ± 0.19	63.47 ± 2.24	70.80 ± 1.84
	HAICS	94.89 ± 0.20	41.29 ± 0.91	88.47 ± 0.53	66.23 ± 1.00	72.95 ± 0.87
	RES-G	**95.91 ± 0.31**	**45.97 ± 0.12**	87.54 ± 0.30	82.88 ± 1.14	82.90 ± 0.33
	RES-L	95.53 ± 0.54	44.64 ± 0.31	86.37 ± 0.08	**88.01 ± 0.39**	**84.78 ± 0.29**

tion kernel as the imputation function achieved the best performance, outperforming the baseline DNN model by 4% and 3% on prediction accuracy on gender classification and scene recognition datasets, respectively. In addition, the proposed RES framework enhanced the explainability of the backbone DNNs by a significant margin as compared with the baseline DNN model as well as other explanation supervision methods. The proposed RES-L with learnable kernels as the imputation function achieved the biggest improvement on model explainability in terms of both IoU and F1 scores on both datasets, outperforming other comparison methods by 8%-72% and 16%-36% on IoU and explanation F1 scores, respectively. The comparison methods GRADIA and HAICS also improved the model performance by leveraging the additional human attention labels, but are generally much less effective than the proposed RES framework. Those results demonstrated the effectiveness of the proposed framework on enhancing the model explainability robustly under noisy annotation labels, and consequently improved the model performance and prediction power on the prediction tasks.

Next, we further studied how the DNN models can benefit from the RES framework to gain a better generalization power under different training sample size scenarios. Specifically, we studied four training sample scenarios with training sample

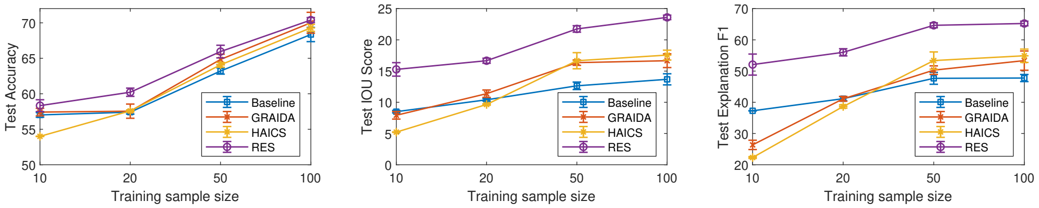


Figure 4.8: Model performance under different training sample size scenarios on gender classification dataset. The data point represents the mean value over 5 runs, and the error bar here corresponds to the standard deviation. (Left) The test prediction accuracy comparison. (Middle) The test IoU score comparison. (Right) The test explanation F1 score comparison.

sizes of 10, 20, 50, and 100 on the Gender Classification Dataset. As shown in Figure 4.8, we present the test prediction accuracy, IoU score, and explanation F1 score of each method under the four training sample size scenarios. The data point represents the mean value over 5 runs, and the error bar here corresponds to the standard deviation. We can see that the proposed RES framework outperformed all other comparison methods by a significant margin under all scenarios studied, especially on boosting the explainability of the backbone DNNs as reflected by IoU and explanation F1 scores. Specifically, RES was able to improve the model prediction accuracy by 2% - 5%, and boosted the quality of the model explanation by 60%-80% and 36%-40% in terms of IoU and explanation F1 scores, respectively. Interestingly, we also observed degradation in model performance when applying GRADIA and HAICS when the sample size is extremely limited, such as in 10 and 20 training sample sizes scenarios. This could be due to the fact that GRADIA and HAICS simply treat the raw human annotation as clear data and thus suffer significantly from learning directly from the noisy labels and consequently prone to over-fitting badly. In contrast, with the robust learning objective, the proposed RES framework was able to cope with the noisy label pretty well even under a very limited sample size, and consequently boosted the model performance in terms of prediction power as well as explainability robustly in all scenarios studied.

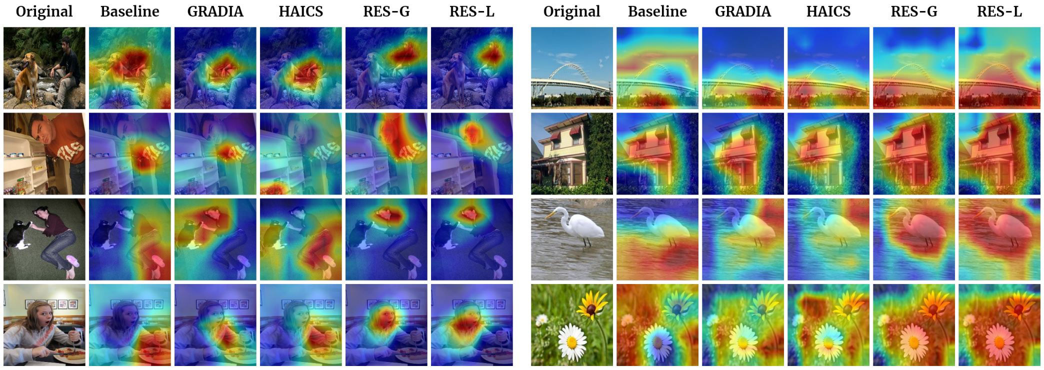


Figure 4.9: Selected explanation visualization results on gender classification dataset (left) and scene recognition dataset (right). The model-generated explanations are represented by the heatmaps overlaid on the original image samples, where more importance is given to the area with a warmer color.

Qualitative Analysis of the Explanation

Case Studies: Here we provide some case studies about the model-generated explanation comparison for both gender classification and scene recognition datasets, as illustrated in Figure 4.9. Here we present the model-generated explanations as the heatmaps overlaid on the original image samples, where more importance is given to the area with a warmer color.

Gender Classification: As shown in the left four rows of Figure 4.9, we studied two 'male' class instances (top 2 rows) and two 'female' class instances (bottom 2 rows). As can be seen, in general, the explanation generated by the proposed RES models can more accurately focus on the important areas (e.g., the human face areas) for identifying the gender of the person in the image. In contrast, both the baseline model as well as the two comparison methods failed to generate reasonable explanation, as the models' 'attention' was distracted by some other objects presented in the images that are irrelevant to the gender classification task. For example, as shown in the first row on the left in Figure 4.9, where both a dog and a person are presented in the image sample. The explanation generated by the baseline and comparison methods assigned importance to the areas in between the dog and the person, therefore, it could not focus properly on the person. On the other hand, both RES-G and RES-L learned to focus only on the person, more specifically on the facial area. Similar patterns could also be observed in the rest three rows on the left, demonstrating the powerful effect of the proposed RES framework on learning to generate more accurate explanations, and consequently enhance the explainability of the DNN models.

Scene Recognition: For the scene recognition dataset, as shown in the right four rows in Figure 4.9, we studied two instances of 'urban' scene (top 2 rows) and two instances of 'nature' scene (bottom 2 rows). Once again, we found that compared with the baseline model and other comparison methods, the explanations generated by RES models are more accurate and close to the ground truth for identifying whether the scene is taken from the urban areas or wild nature. For instance, as shown in the third row on the right in Figure 4.9, the explanation generated by both the baseline and comparison methods focuses more on the water surface while RES focuses more on the wild animal itself. Similarly, as shown in the fourth row, the explanation generated by RES focuses more on the wildflowers than the grass-field background. Although in those situations the prediction can be correct for all the models studied, we argue that the model trained with the RES framework can be more robust and have a batter generalizability power to the downstream predictive tasks by learning to assign importance more accurately to the most distinguishable features/patterns presented in the data samples.

Human Assessment: To evaluate the quality of explanations for the five comparison methods, we developed a web-based user interface (UI) where a human annotator can go over all the model-generated explanations and make qualitative evaluation on both datasets. We distributed the model-generated explanations from the test set to three separate human annotators. We asked annotators to assess the perceived quality of explanations with the five-level Likert scale. "5-Excellent" when explana-

Model Pairs	Perceived Quality (p-values)
Baseline vs. GRADIA	**2.68e-03**‡
Baseline vs. HAICS	**2.33e-04**‡‡
Baseline vs. RES-G	**4.98e-37**‡‡
Baseline vs. RES-L	**4.96e-28**‡‡
GRADIA vs. HAICS	0.4980
GRADIA vs. RES-G	**2.71e-22**‡‡
GRADIA vs. RES-L	**1.54e-15**‡‡
HAICS vs. RES-G	**1.67e-19**‡‡
HAICS vs. RES-L	**2.96e-13**‡‡
RES-G vs. RES-L	0.0824

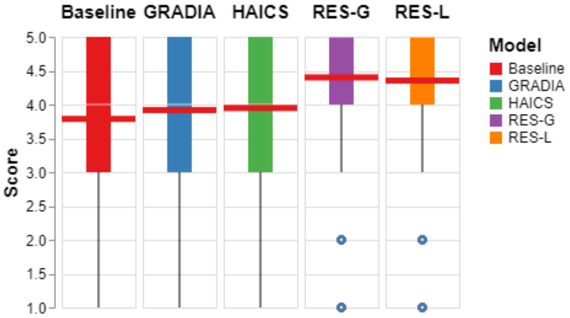


Figure 4.10: Top: results for pairwise comparison of five conditions. †: $p < 0.05$, ‡: $p < 0.01$, ‡‡: $p < 0.001$. Bottom: Distributions of human users' perceived attention quality ratings. 5-level Likert scale is used (5: Excellent, 4: Good, 3: Fair, 2: Bad, 1: Inferior).

tions show positive attention very clearly while don't contain negative attention at all, and "4-Good" when positive attention is clearly presented with negligible negative attention. "3-Fair" meant that positive attention is partially seen while negative attention is clearly visible. "2-Bad" in case positive attention can be barely seen while negative can be found evidently. "1-Inferior" is assigned when a human annotator can only find negative attention. After performing the Shapiro-Wilk normality test, we found participants' ratings don't follow a normal distribution. Therefore, we applied Kruskal-Wallis H-test for identifying the differences between the five conditions. The quality ratings of five models are significantly different, with a p-value of 7.82e-51 (¡ 0.05). For post-hoc pairwise comparisons using Dunn's test, all pairs are significantly different, with the exception of GRADIA vs. HAICS and RES-G vs. RES-L. This means that the ranking among the five conditions is that RES-G (M = 4.40, SD = 0.91) and RES-L (M = 4.35, SD = 0.89) are rated notably higher than

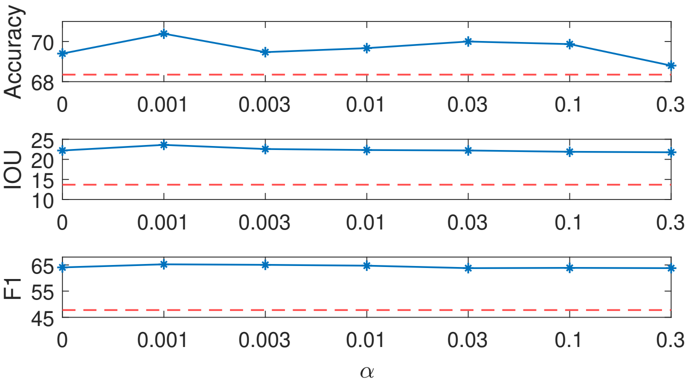


Figure 4.11: The sensitivity study of hyper-parameter α in RES framework (RES-L) on gender classification dataset. The red dashed lines represent the baseline model's performance.

the rest, followed by GRADIA (M = 3.92, SD = 1.24) and HAICS (M = 3.95, SD = 1.23). The least performing condition was Baseline (M = 3.79, SD = 1.25). Specific pair-wise testing results and visual representation between conditions are shown in Figure 4.10.

Sensitivity Analysis of Hyper-parameter

Here we further provide a sensitivity analysis of the hyper-parameter α introduced in the proposed RES framework, as shown in Equation (4.22) which measures the tolerance level we give to the discrepancies between human annotation labels and the model explanation. Figure 4.11 shows the prediction accuracy, IoU, and explanation F1-score of the RES-L model for various values of α on the gender classification dataset. The scene recognition dataset follows a similar trend. The red dashed lines represent the baseline model's performance. In general, the model performance is not too sensitive to the value of α within the range studied, as all models outperformed the baseline model by a significant margin in terms of both prediction accuracy as well as explainability. As we developed our models based on the accuracy of the validation set, we indeed observed a concave curvature on test accuracy, peaking at a α value between 0.001 and 0.1. While the specific best value of α can vary depending on the dataset as well as the degree of nosiness of the human annotation labels (such as the granularity of the annotation), in general, the proposed framework can perform well when α is relatively small (e.g., less than 0.1).

4.4 Conclusion

We observe several side-effects behind the DNNs' powerful automation as a form of "bias" every day. We are directly or indirectly influenced by the AI's decisions affected by automated racism, gender bias, lack of considering people with a neurodiverse spectrum, insecurities on adversarial attacks, and many more. CSCW, HCI, and broader ML communities have invested substantial effort into devising straightforward and human-usable solutions for effectively aligning DNNs' behavior with our norms and expectation. However, several empirical studies revealed that steering DNNs as we intended is highly challenging not only for domain experts but also for skilled data scientists.

The overarching motivation behind our study here is to devise a human-usable interaction modality that a human can directly see how DNNs think and intuitively modify the cases when needed. To do so, this work aimed at laying the groundwork for establishing a platform that can use EGL framework to more directly infuse their perspectives in fine-tuning DNNs. To this end, we propose GRADIA and RES as generic EGL frameworks for visual explanation-guided learning by developing novel explanation model objectives that can handle the noisy human annotation labels as the supervision signal with a theoretical justification of the benefit to model generalizability. Extensive experiments on two real-world image datasets demonstrate the effectiveness of the proposed framework on enhancing both the reasonability of the explanation as well as the performance of the backbone DNNs model. Although the additional data of human explanation labels may not be easily accessible, our studies have demonstrated the effectiveness of the proposed framework under a quite limited amount of training samples, which could benefit application domains where data samples are limited and hard to acquire, yet both model performance as well as the explainability are on-demand, such as in medical domains.

As a closing remark, we hope this work can motivate future research in EGL on DNNs and more generally devising novel interaction modalities that can realize DNNs that better align with a human mental model.

Chapter 5

Conclusions and Future Works

This dissertation aims to handle two main problems for geometric data: how to enhance the interpretability of geometric neural networks, including CNNs and GNNs, and how the explanations can further help improve the model in terms of generalizability. For enhancing the interpretability of geometric DNNs, we explore three sub-tasks, namely graph-structured multi-task representation learning for event forecasting, interpretable and efficient bio-inspired deep learning via neuronal assemblies, and interpretation for dynamic attributed Graphs via hierarchical attention. To further explore how to improve the model in terms of both interpretability and generalizability, we explore two sub-tasks, namely, the explanation-guided representation learning on image as well as graph-structured data.

To study how to enhance the interpretability of geometric data, we first explored the bio-inspired neuronal assemblies to help making the model more intrinsically interpretable and efficient. In this work, we propose a novel Biologically Enhanced Artificial Neuronal assembly (BEAN) regularization to model neuronal correlations and dependencies inspired by cell assembly theory from neuroscience. We show that BEAN can promote jointly sparse and efficient encoding of rich semantic correlation among neurons in DNNs similar to connection patterns in BNNs. Experimental results show that BEAN enables the formations of interpretable neuronal functional clusters and consequently promotes a sparse, memory/computation-efficient network without loss of model performance. Moreover, our few-shot learning experiments demonstrated that BEAN could also enhance the generalizability of the model when training samples are extremely limited. Our regularization method has demonstrated its capability in enhancing the modularity of the representations of neurons for image semantic meanings such as digits, animals, and objects on image datasets.

Next, we extended the knowledge we gained from general DNNs to GNNs, by exploring the interpretability for dynamic attributed graphs on online health forum data. Specifically, we formulated the task of health stage inference using online health forum data as a dynamic graph-to-sequence learning problem and propose a novel dynamic graph-to-sequence neural networks architecture (DynGraph2Seq) that can handle this new type of learning problem effectively. Our DynGraph2Seq model consists of a novel dynamic graph encoder and an interpretable sequence decoder to learn the mapping between a sequence of time-evolving user activity graphs and a sequence of target health stages. In addition, we developed a new dynamic graph regularization and dynamic graph hierarchical attention to facilitate the multi-level interpretability. Our comprehensive experiments and analyses for health stage prediction demonstrate both the effectiveness and the interpretability of the proposed models.

To further explore how to improve the model in terms of both interpretability and generalizability, we leveraged the explanation-guided learning techniques that can learn to explain DNNs on geometric data, including image and graph-structured data. For image data, we propose GRADIA and RES frameworks for visual explanation supervision by developing a novel explanation model objectives that can handle the noisy human annotation labels as the supervision signal with a theoretical justification of the benefit to model generalizability. Extensive experiments on two real-world image datasets demonstrate the effectiveness of the proposed framework on enhancing both the reasonability of the explanation as well as the performance of the backbone CNNs model. Although the additional data of human explanation labels may not be easily accessible, our studies have demonstrated the effectiveness of the proposed RES framework under a quite limited amount of training samples, which could benefit application domains where data samples are limited and hard to acquire, yet both model performance as well as the explainability are on-demand, such as in medical domains. For graph-structured data, we propose a GNN Explanation Supervision (GNES) framework to adaptively learn how to explain GNNs more correctly. Specifically, our framework jointly optimizes both model prediction and model explanation by enforcing both whole graph regularization and weak supervision on model explanations. For the graph regularization, we propose a unified explanation formulation for both node-level and edge-level explanations by enforcing the consistency between them. The node- and edge-level explanation techniques we propose are also generic and rigorously demonstrated to cover several existing major explainers as special cases. Extensive experiments on five real-world datasets across two application domains demonstrate the effectiveness of the proposed model on improving the reasonability of the explanation while still keep or even improve the backbone GNNs model performance.

Finally, we aim at extend the proposed techniques and jointly work with Human-Computer-Interaction researchers to design some real-world system for supporting the further advancement of the EGL research community. We have worked with the Human-Computer Interaction (HCI) domain researchers from our interdisciplinary team and developed an online interactive tool called ‘DeepFuse’ (currently under review of CHI 2023) that enables the very first end-to-end tool for machine learning practitioners to explore EGL training with their own datasets and machine learning models of interest.

5.1 Research Tasks

The major research tasks are described as follows. The current status of these tasks is listed in Table 5.1.

5.1.1 Development of Interpretability Techniques for DNNs

- **Proposal of the BEAN regularization (A1)**. we propose a Biologically Enhanced Artificial Neuronal assembly (BEAN) regularization that promoting jointly sparse and efficient encoding of rich semantic correlation among neurons, and enhancing model generalizability with few training samples.

- **Validation on the model efficiency (A2)**. we conducted efficiency test of the trained model to show that BEAN enables the formation of interpretable neuronal functional clusters and consequently promotes a sparse, memory/computation-efficient network without loss of model performance.

- **Validation on few-shot learning (A3)**. We performed few-shot learning experiments and demonstrate that BEAN could also enhance the generalizability of the model when training samples are extremely limited.

- **Proposal of the DynGraph2Seq framework (A4)**. We defined the novel problem of inferring user health stage information using online health forum data and proposed a generic framework DynGraph2Seq for inferring target sequence from a sequence of graphs.

- **Validation on the interpretability (A5)**. We proposed a dynamic graph regularization that enforces the smooth learning of consecutive graphs while preserving the heterogeneity across the graph sequence. In addition, we propose a new dynamic graph hierarchical attention mechanism that captures both the time-level and node-level attention, thus providing model transparency throughout the whole inference process.

5.1.2 Explanation-Guided Learning on Graphs

- **Proposal of interactive Explanation-Guided Learning framework for GNNs (B1).** We present a new learning objective for joint optimization among the model prediction loss, the explanation loss, and the graph regularization loss on regulating the model explanation. In addition, our framework can treat the explanation loss as an optional term and thus work effectively in scenarios where the human annotation on explanation is limited.
- **Development of unified graph-based explanation frameworks node- and edge-level explanation (B2).** We proposed a unified EGL framework for both node-level and edge-level explanations that is suitable for explanation supervision and generalizable to the existing differentiable explanation methods.
- **Proposal of novel node- and edge-level explanation regularization (B3).** We propose to apply novel explanation regularizations (i.e., explanation consistency and sparsity) onto the model-generated explanation to inject general graph principles and prior knowledge about the explanation that enhance the quality and consistency among the multiple levels of explanations.
- **Validation on real-world datasets (B4).** Extensive experiments on five real-world datasets in two domains, chemical (molecular graphs) and vision (scene graphs), demonstrate that the proposed models improved the backbone GNN model both in terms of prediction power and explainability across different application domains. In addition, qualitative analyses, including case studies and user studies of the model explanation, are provided to demonstrate the effectiveness of the proposed framework.

5.1.3 Explanation-Guided Learning on Images

- **Proposal of interactive Explanation-Guided Learning framework for CNNs (C1)**. We propose a novel EGL framework that leverages reasonability matrix to (1) systematically detect biased reasoning and (2) effectively remove it through a direct human intervention. We present GRADIA, a novel technique that strikes the balance between prediction accuracy and attention accuracy in fine-tuning DNNs.

- **Proposal of robust Explanation-Guided Learning framework under noisy annotation labels (C2)**. We propose a unified EGL framework that enables explanation supervision on DNNs with both positive and negative explanation annotation labels and is generalizable to the existing differentiable explanation methods. We propose a novel robust explanation loss that can handle the inaccurate boundary, incomplete region, as well as inconsistent distribution challenges in applying the noisy human annotation labels as the supervision signal.

- **Validation on real-world image datasets (C3)**. Extensive experiments on two real-world image datasets, gender classification and scene recognition, demonstrate that the proposed framework improved the backbone DNNs both in terms of prediction power and explainability.

- **Validation with user studies and qualitative human evaluations (C4)**. We also conducted qualitative analyses, including case studies and user studies of the model explanation, are provided to demonstrate the effectiveness of the proposed framework.

- **Build up the interactive EGL tools for real world applications (C5)**. We work with the Human-Computer Interaction (HCI) domain experts to form an interdisciplinary team and develop an online interactive tool called 'DeepFuse' (currently under review of CHI 2023) that enables the very first end-to-end tool for machine learning practitioners to explore EGL training with their own datasets and machine learning models of interest.

Table 5.1: Research tasks and status

Task	Description	Status
Research Area A	**Development of Interpretability Techniques for DNNs**	
A1	Proposal of the BEAN regularization	Completed
A2	Validation on the model efficiency	Completed
A3	Validation on few-shot learning	Completed
A4	Proposal of the DynGraph2Seq framework	Completed
A5	Validation on the Interpretability	Completed
Research Area B	**Explanation-Guided Learning on Graphs**	
B1	Proposal of interactive Explanation-Guided Learning framework for GNNs	Completed
B2	Development of unified graph-based explanation frameworks node- and edge-level explanation	Completed
B3	Proposal of novel node- and edge-level explanation regularization	Completed
B4	Validation on real-world datasets	Completed
Research Area C	**Explanation-Guided Learning on Images**	
C1	Proposal of interactive Explanation-Guided Learning framework for CNNs	Completed
C2	Proposal of robust Explanation-Guided Learning framework under noisy annotation labels	Completed
C3	Validation on real-world image datasets	Completed
C4	Validation with user studies and qualitative human evaluations	Completed
C5	Build up the interactive EGL tools for real world applications	Completed
D	Dissertation Writing and revision	Completed

5.2 Publications

5.2.1 Published papers

- **Yuyang Gao**, Tong Steven Sun, Sungsoo Ray Hong, and Liang Zhao. Aligning Eyes between Humans and Deep Neural Network through Interactive Attention Alignment. Proceedings of the ACM on Human-Computer Interaction (CSCW 2022).

- **Yuyang Gao**, Tong Steven Sun, Guangxi Bai, Siyi Gu, Sungsoo Ray Hong, and Liang Zhao. RES: A Robust Framework for Guiding Visual Explanation. The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2022).

- **Yuyang Gao**, Tong Sun, Rishab Bhatt, Dazhou Yu, Sungsoo Hong, and Liang Zhao. GNES: Learning to Explain Graph Neural Networks. The 21st IEEE International Conference on Data Mining (ICDM 2021).

- **Yuyang Gao**, Giorgio Ascoli, Liang Zhao. Schematic Memory Persistence and Transience for Efficient and Robust Continual Learning. Neural Networks, 144 (2021) 49–60.

- **Yuyang Gao**, Tanmoy Chowdhury (co-first author), Lingfei Wu, Liang Zhao. Modeling Health Stage Development of Patients with Dynamic Attributed Graphs in Online Health Communities. IEEE Transactions on Knowledge and Data Engineerings (TKDE), 2021.

- **Yuyang Gao**, Giorgio Ascoli, Liang Zhao. BEAN: Interpretable and Efficient Learning with Biologically-Enhanced Artificial Neuronal Assembly. Frontiers in Neurorobotics, 2021.

- **Yuyang Gao**, Lingfei Wu, Houman Homayoun, and Liang Zhao. DynGraph2Seq: Dynamic-Graph-to-Sequence Interpretable Learning for Health Stage Prediction in Online Health Forums. The 19th International Conference on Data Mining (ICDM 2019), Beijing, China, Nov 2019.

- **Yuyang Gao**, Liang Zhao, Lingfei Wu, Yanfang Ye, Hui Xiong, Chaowei Yang. Incomplete Label Multi-task Deep Learning for Spatio-temporal Event Subtype Forecasting. Thirty-third AAAI Conference on Artificial Intelligence (AAAI 2019), Hawaii, USA, Feb 2019.

- **Yuyang Gao**, Xiaojie Guo, Liang Zhao. Local Event Forecasting and Synthesis Using Unpaired Deep Graph Translations. 2nd ACM SIGSPATIAL International Workshop on Analytics for Local Events and News (LENS 2018), Seattle, Washington, USA, Nov 2018.

- **Yuyang Gao** and Liang Zhao. Incomplete Label Multi-Task Ordinal Regression for Spatial Event Scale Forecasting. Thirty-Second AAAI Conference on Artificial Intelligence (AAAI 2018), New Orleans, US, Feb 2018.

- Negar Etemadyrad, **Yuyang Gao**, Qingzhe Li, Xiaojie Guo, Frank Krueger, Qixiang Lin, Deqiang Qiu, and Liang Zhao. 2022. Functional Connectivity Prediction with Deep Learning for Graph Transformation. IEEE Transactions on Neural Networks and Learning Systems (TNNLS).

- Liang Zhao, **Yuyang Gao**, Jieping Ye, Feng Chen, Fanny Ye, Chang-tien Lu, and Naren Ramakrishnan. Spatio-temporal Event Forecasting Using Incremental Multi-source Feature Learning. ACM Transactions on Knowledge Discovery from Data (TKDD), 2021.

- Junxiang Wang, **Yuyang Gao**, Andreas Zufle, Jingyuan Yang, and Liang Zhao. Incomplete Label Uncertainty Estimation for Petition Victory Prediction with Dynamic Features. In Proceedings of the IEEE International Conference on Data Mining (ICDM 2018), Singapore, Nov 2018.

5.2.2 Submitted and In-preparation papers

- **Yuyang Gao**, Siyi Gu, Junji Jiang, Sungsoo Ray Hong, Dazhou Yu, and Liang Zhao. Beyond XAI: A Systematic Survey for Explanation-Guided Learning. ACM Computing Surveys (CSUR), submitted.

- **Yuyang Gao**, Junxiang Wang, Wei Wang, Xin Deng, Hamed Zamani, Xiaohan Yan, Yan Guo, Ahmed Awadallah, Yanfang Ye, and Liang Zhao. Asynchronous Semi-supervised Representation Learning for Email Heterogeneous Networks. In-preparation.

- Tong Steven Sun, **Yuyang Gao**, Shubham Khaladkar, Sijia Liu, Liang Zhao, Young-Ho Kim, and Sungsoo Ray Hong. DeepFuse: Making Convolutional Neural Networks "Think Like Humans" through Case-based Vulnerability Detection and Model Steering. ACM CHI Conference (CHI 2023), under review.

- Nahyun Kwon, Tong Steven Sun, **Yuyang Gao**, Liang Zhao, Xu Wang, Jeeeun Kim, and Sungsoo Ray Hong. 3DPFIX: Assisting Remote Novices' 3D Printing Troubleshooting through Community-Minded Human-AI Collaboration Design. In-preparation.
- Guangji Bai, Chen Ling, **Yuyang Gao**, and Liang Zhao. Saliency-augmented Memory Completion for Continual Learning. Eleventh International Conference on Learning Representations (ICLR 2023), submitted.

5.3 Future Research Directions

5.3.1 Explanation-Guided Learning on Medical Image Analysis

Besides generic image applications, Explanation-Guided Learning has also been widely studied in the medical domain, thanks to the availability of domain-expert annotation on many medical image datasets [29, 76, 154]. In general, we observed a variety of datasets studied by existing works, including but not limited to ISIC Skin Cancer dataset [29], Iris-Cancer dataset [84], scaphoid fracture detection dataset [76], Fundus image dataset (IDRiD) [109], and the pneumonia detection X-ray dataset [154] for disease identification task [196]. Similar to most EGL frameworks on generic image data, an additional explanation loss is added to the model objective and is typically realized by a distance loss between the ground truth annotation collected from domain experts and the model visual explanation. However, compared with generic image data, several unique challenges have been identified by existing works when applying EGL to medical images, such as 1) difficulty in assessing the quality of the model explanation, and 2) the scalability of the sample size of the annotation labels of the datasets.

5.3.2 Trustworthiness and Fairness of Deep Learning Explanation

Fairness, Accountability, and Transparency (FaccT) are becoming as important as–or depending on application areas–more important than model accuracy as an evaluation metric. Since it is nearly not feasible to prepare an impeccable dataset that can equally represent every possible feature related to a model's task, blindly pursuing a model's accuracy cannot exclude the chance of causing "catastrophic consequences" in critical circumstances [61]. One of EGL's crucial application areas is to realize the balance between the model accuracy and FaccT by allowing human users to elicit their perspectives on steering the model. In shaping the balance, one crucial research direction is to understand how to maximize the case where reasonable human reasoning can also cause accurate prediction. There are several arguments discussing when human reasoning can cause a beneficial or detrimental effect on model prediction. While the debate is ongoing, we are gradually seeing more evidence where human involvement can result in a positive effect [49, 30]. For example, Shao et al. find humans "arguing against" unreasonable explanation can benefit the model [132]. At the end of the day, from the perspective of model accuracy and FaccT, a railroad should not the reason for predicting a train [80], a snowboard cannot be a male class [58], and a shopping cart should not only belong to a woman class [186].

5.3.3 Contrastive Explanation-Guided Learning

Contrastive learning is a powerful self-supervised learning strategy that encourages augmentations of the same input to have more similar representations compared to augmentations of different inputs. In the field of EGL, we have started to see several works that apply the contrastive objective to the model explanation between similar/dissimilar samples to build up the explanation objective [176, 35, 136, 106]. The most significant advantage of leveraging the contrastive learning paradigm for explanation guidance is that no ground truth explanation annotation labels are required for model training. However, designing an appropriate contrastive framework for EGL can be more challenging due to the lack of a standard form of model explanation under different application domains. Besides, how to define and formulate the positive and negative explanation samples to contrast with the anchor sample's explanation can be challenging without knowing the ground-truth labels. Thus, we believe the further development of the contrastive EGL framework can be one of the core future directions in EGL, and it can lead to a significant leap in the application of EGL to the domains where ground truth explanation labels are generally difficult to obtain in large scale.

5.3.4 Interactive Explanation-Guided Learning pipeline on Continual & Active Learning

EGL's core principle is motivating ML engineers' iterative training, such as continual learning [125, 36] and active learning [67, 21]; helping them to figure out the vulnerability through explanation and fixing the issue by providing a human-level guideline. In supporting such an iterative training, we believe one of the promising areas is "data iteration", a design that can help ML engineers to fortify the dataset by adding more examples based on detected vulnerabilities through explanation. In such a direction, we believe understanding the pros and cons of retraining and continual learning can be crucial. For example, there can be a case where newly found data points can be stacked up on an existing dataset and be used in retraining. Another case can be to iteratively update the last model through some of the existing techniques in continual learning [104]. In general, in the world of EGL, understanding when to apply retraining or continual learning and what are the pros and cons of each training strategy are not well understood. Understanding which strategy can yield what strengths and weaknesses in the scenario of data iteration would be one of the core future applications of EGL.

Appendix A

Explanation-Guided Representation Learning on Geometric Data

A.1 Proof of Theorem 2

Proof. Suppose f_ψ is a δ-minimizer of $\mathcal{L}$ with $\psi \in \Theta$. From Assumption 1, we know that there exists a neural network f_τ such that

$$\|f_\tau - f_{\theta^*}\|^2 \coloneqq \mathbb{E}\left[|f_\tau - f_{\theta^*}|^2 + |\nabla f_\tau - \nabla f_{\theta^*}|^2\right] \leq C_1^2 \frac{\|\theta^*\|^2}{m^\gamma} \tag{A.1}$$

Then, we have

$$\begin{aligned} \mathcal{L}(f_\psi) - \mathcal{L}(f_{\theta^*}) &\leq \mathcal{L}(f_\tau) - \mathcal{L}(f_{\theta^*}) + \delta \\ &\leq L_1 \mathbb{E}\left[|f_\tau(x) - f_{\theta^*}(x)\right] + L_2 \mathbb{E}\left[|\nabla f_\tau(x) - \nabla f_{\theta^*}(x)\right] + \delta \\ &\leq (L_1 + L_2) \frac{C_1 \|\theta^*\|}{m^{\gamma/2}} + \delta \end{aligned} \tag{A.2}$$

From Assumption 2, given $0 < \epsilon < 1$, we have

$$P(|\mathcal{L}(f_\theta) - \hat{\mathcal{L}}(f_\theta)| \leq \frac{C_2(V, m, \epsilon)}{\sqrt{N}}) \geq 1 - \epsilon, \quad \forall\, \theta \in \Theta \tag{A.3}$$

Then,

$$\begin{aligned}
\mathcal{L}(f_{\hat{\theta}}) - \mathcal{L}(f_{\theta^*}) &\leq \hat{\mathcal{L}}(f_{\hat{\theta}}) - \mathcal{L}(f_{\theta^*}) + \frac{C_2(V, m, \epsilon)}{\sqrt{N}} \\
&\leq \hat{\mathcal{L}}(f_{\psi}) - \mathcal{L}(f_{\theta^*}) + \frac{C_2(V, m, \epsilon)}{\sqrt{N}} + \delta \\
&\leq \mathcal{L}(f_{\psi}) - \mathcal{L}(f_{\theta^*}) + \frac{C_2(V, m, \epsilon)}{\sqrt{N}} + \delta \\
&\leq (L_1 + L_2)\frac{C_1\|\theta^*\|}{m^{\gamma/2}} + \frac{2C_2(V, m, \epsilon)}{\sqrt{N}} + 2\delta
\end{aligned} \tag{A.4}$$

□

A.2 Proof of Lemma 1

Proof. Since

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{1 - e^{-2x}}{1 + e^{-2x}} \tag{A.5}$$

where the last equality follows by multiplying by $\frac{e^{-x}}{e^{-x}} = 1$. And since: $\lim_{x\to\infty} 1 - e^{-2x} = 1$, and $\lim_{x\to\infty} 1 + e^{-2x} = 1$, we have

$$\lim_{x\to\infty} tanh(x) = 1 \tag{A.6}$$

Similarly, we also have

$$\lim_{x\to-\infty} tanh(x) = \lim_{x\to-\infty} \frac{e^{2x} - 1}{e^{2x} + 1} = -1 \tag{A.7}$$

Thus we have

$$\lim_{\gamma \to \infty} tanh(\gamma(M^{(i)} - a)) = \begin{cases} 1 & M^{(i)} > a \\ -1 & M^{(i)} < a \end{cases} \tag{A.8}$$

Thus we have the equivalency of Equation (4.25) and Equation (4.22) when $\gamma \to \infty$. □

A.3 Human Annotation and Evaluation UI demonstration

Figure A.1 (a) is the interface used to collect attention annotation on the areas people think are relevant to the classification task. For example, for the gender dataset annotation, users first determine whether they can identify the person's gender in the image, then draw the areas that help them for the gender classification. In the back-end, the coordinates of highlighted areas are converted into a binary map, preparing for the modeling step.

Figure A.1 (b) is the interface for human assessment on the model-generated explanations. For each image annotation, 5 explanations were presented in random order with 3 questions (Q1 and Q2 are true/false questions, Q3 is a 5-point Likert scale rating question) asked for each explanation. Question 1 asks if the focus on the explanation shows details necessary for identifying the target label (i.e., labels in gender classification or scene recognition), and question 2 asks for the presence of unnecessary details on the image for identifying the target. Question 3 is our main focus of the attention quality assessment, where annotators give 1 to 5 ratings to each model explanation.

A.4 Efficient Adaptive Threshold Searching Algorithm

Algorithm 2: Adaptive Threshold Searching Algorithm

```
Require: $M, F, C$
Ensure: solution $a$
initialize: $a = 0, act = 0, v = 0, vct = 0, i = 0, j = 0$
$ge = \{M[find(C > 0)]\}$ % find the set of greater or equal to inequality constraints
$l = \{M[find(F > 0)]\}$ % find the set of less to inequality constraints
$ges = \mathrm{Sort}(ge, \text{‘ascend’})$
$ls = \mathrm{Sort}(l, \text{‘descend’})$
for $i < |ges|$ do
    $v = ges[i]$
    $vct = i + 1 + \mathrm{BinarySearch}(v, ls)$
    if $vct > act$ then
        $a = v$
        $act = vct$
    end if
    $i = i + 1$
end for
for $j < |ls|$ do
    $v = ls[i]$
    $vct = j + 1 + \mathrm{BinarySearch}(v, ges)$
    if $vct > act$ then
        $a = v$
        $act = vct$
    end if
    $j = j + 1$
end for
```

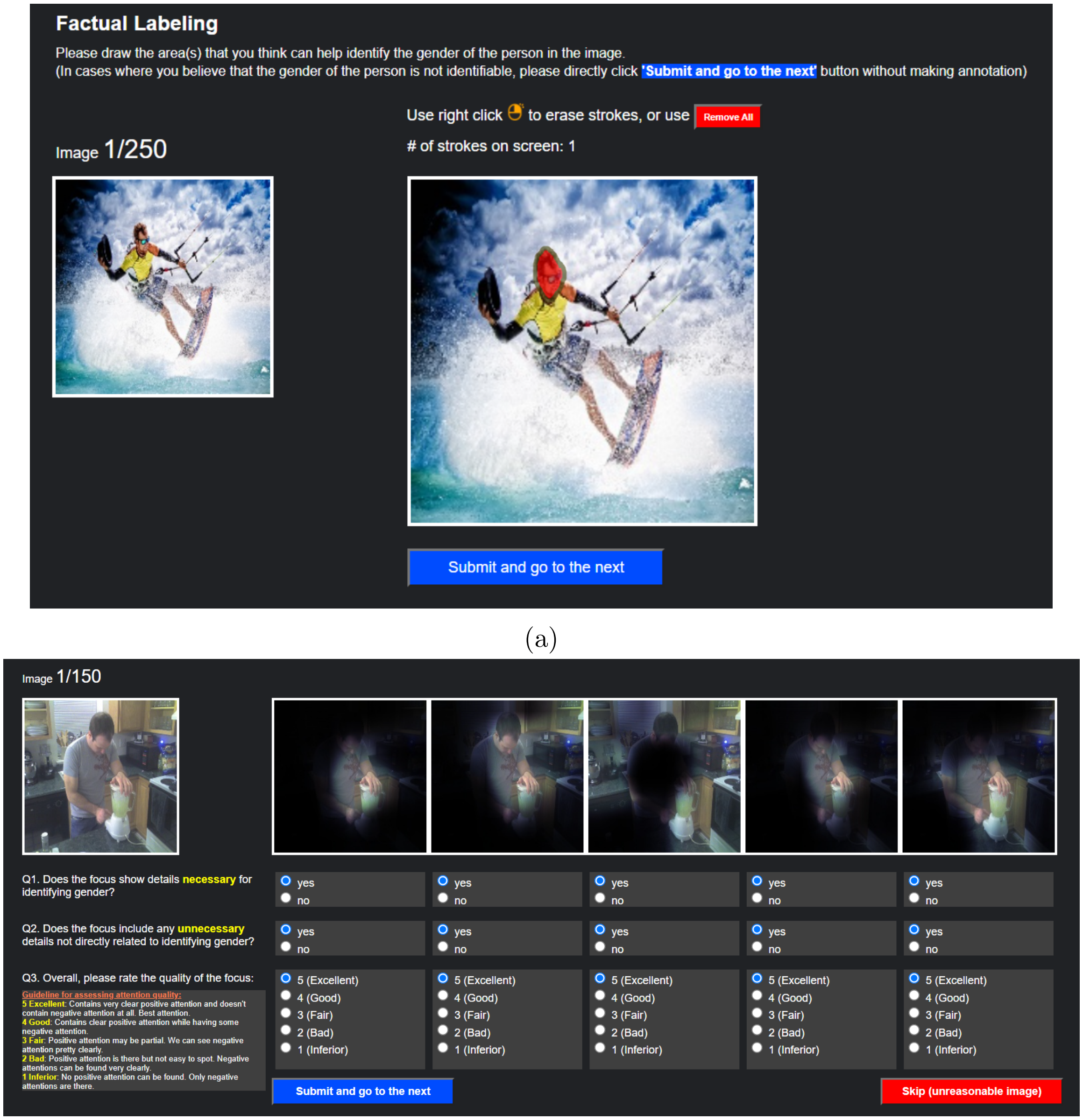
A.5 Detailed Implementation of the Learnable Imputation Layers

For the learnable imputation function, we studied both a shallow implementation as well as a deep implementation, as shown in detail below:

Shallow Implementation: We apply one layer of convolution operation to process the raw human annotation label, with a 64×64 convolution kernel with a padding size of 16 and a stride of 32.

Deep Implementation: We apply five layers of convolution operations to process the raw human annotation label, with 7×7, 3×3, 3×3, 3×3, and 3×3 convolution kernel with a padding size of 3 on the first layer and 1 for the rest layer, and a stride 2 for all layers.

We choose the Shallow implementation for the RES-L model as it achieves better performance on the validation set. The reason why the deep version gets inferior performance could be due to the training sample size studied in this work is too small.



Factual Labeling

Please draw the area(s) that you think can help identify the gender of the person in the image.
(In cases where you believe that the gender of the person is not identifiable, please directly click **'Submit and go to the next'** button without making annotation)

Image 1/250

Use right click to erase strokes, or use

# of strokes on screen: 1

(a)

Image 1/150

Q1. Does the focus show details **necessary** for identifying gender?	◉ yes ○ no	◉ yes ○ no	◉ yes ○ no	◉ yes ○ no	◉ yes ○ no
Q2. Does the focus include any **unnecessary** details not directly related to identifying gender?	◉ yes ○ no	◉ yes ○ no	◉ yes ○ no	◉ yes ○ no	◉ yes ○ no
Q3. Overall, please rate the quality of the focus:	◉ 5 (Excellent) ○ 4 (Good) ○ 3 (Fair) ○ 2 (Bad) ○ 1 (Inferior)	◉ 5 (Excellent) ○ 4 (Good) ○ 3 (Fair) ○ 2 (Bad) ○ 1 (Inferior)	◉ 5 (Excellent) ○ 4 (Good) ○ 3 (Fair) ○ 2 (Bad) ○ 1 (Inferior)	◉ 5 (Excellent) ○ 4 (Good) ○ 3 (Fair) ○ 2 (Bad) ○ 1 (Inferior)	◉ 5 (Excellent) ○ 4 (Good) ○ 3 (Fair) ○ 2 (Bad) ○ 1 (Inferior)

Guideline for assessing attention quality:
5 Excellent: Contains very clear positive attention and doesn't contain negative attention at all. Best attention.
4 Good: Contains clear positive attention while having some negative attention.
3 Fair: Positive attention may be partial. We can see negative attention pretty clearly.
2 Bad: Positive attention is there but not easy to spot. Negative attentions can be found very clearly.
1 Inferior: No positive attention can be found. Only negative attentions are there.

(b)

Figure A.1: The screenshots illustrating the two UIs for human annotation and evaluation. (a) The interface for attention annotation where users can draw on the image and generate a binary matrix of the focus area used for improving model explanation quality. (b) The interface for attention quality assessment where 5 model-generated explanations are displayed in random order. Users will answer three questions for each explanation.

Bibliography

[1] American cancer society. `http://www.cancer.org`.

[2] Breast cancer community. `https://community.breastcancer.org/`.

[3] ehealth forum. `http://ehealthforum.com`.

[4] Sami Abu-El-Haija, Bryan Perozzi, Rami Al-Rfou, and Alex Alemi. Watch your step: Learning graph embeddings through attention. *arXiv preprint arXiv:1710.09599*, 2017.

[5] Amina Adadi and Mohammed Berrada. Peeking inside the black-box: a survey on explainable artificial intelligence (xai). *IEEE access*, 6:52138–52160, 2018.

[6] Jose M Alvarez and Mathieu Salzmann. Learning the number of neurons in deep networks. In *Advances in Neural Information Processing Systems*, pages 2270–2278, 2016.

[7] KM Annervaz, Somnath Basu Roy Chowdhury, and Ambedkar Dukkipati. Learning beyond datasets: Knowledge graph augmented neural networks for natural language processing. *arXiv preprint arXiv:1802.05930*, 2018.

[8] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence

(xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115, 2020.

[9] Giorgio A Ascoli and John C Atkeson. Incorporating anatomically realistic cellular-level connectivity in neural network models of the rat hippocampus. *Biosystems*, 79(1-3):173–181, 2005.

[10] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7): e0130140, 2015.

[11] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[12] Guangji Bai and Liang Zhao. Saliency-regularized deep multi-task learning. In *Proceedings of the 28th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2022.

[13] Federico Baldassarre and Hossein Azizpour. Explainability techniques for graph convolutional networks. *arXiv preprint arXiv:1905.13686*, 2019.

[14] Pinar Barlas, Kyriakos Kyriakou, Olivia Guest, Styliani Kleanthous, and Jahna Otterbacher. To" see" is to stereotype: Image tagging algorithms, gender recognition, and the accuracy-fairness trade-off. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW3):1–31, 2021.

[15] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *CVPR*, pages 6541–6549, 2017.

[16] Daniel Beck, Gholamreza Haffari, and Trevor Cohn. Graph-to-sequence learning using gated graph neural networks. *arXiv preprint arXiv:1806.09835*, 2018.

[17] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[18] Mariusz Bojarski, Anna Choromanska, Krzysztof Choromanski, Bernhard Firner, Larry Jackel, Urs Muller, and Karol Zieba. Visualbackprop: visualizing cnns for autonomous driving. *arXiv preprint arXiv:1611.05418*, 2, 2016.

[19] Valentino Braitenberg. Cell assemblies in the cerebral cortex. In *Theoretical approaches to complex systems*, pages 171–188. Springer, 1978.

[20] György Buzsáki. Neural syntax: cell assemblies, synapsembles, and readers. *Neuron*, 68(3):362–385, 2010.

[21] Shiyu Chang, Yang Zhang, Mo Yu, and Tommi Jaakkola. A game theoretic approach to class-wise selective rationalization. *Advances in neural information processing systems*, 32, 2019.

[22] Shi Chen, Ming Jiang, Jinhui Yang, and Qi Zhao. Air: Attention with reasoning capability. In *European Conference on Computer Vision*, pages 91–107. Springer, 2020.

[23] Yu Cheng, Felix X Yu, Rogerio S Feris, Sanjiv Kumar, Alok Choudhary, and Shi-Fu Chang. An exploration of parameter redundancy in deep networks with circulant projections. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2857–2865, 2015.

[24] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase repre-

sentations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[25] Minsuk Choi, Cheonbok Park, Soyoung Yang, Yonggyu Kim, Jaegul Choo, and Sungsoo Ray Hong. Aila: Attentive interactive labeling assistant for document classification through attention-based deep neural networks. In *CHI*, New York, NY, USA, 2019. ACM.

[26] François Chollet et al. Keras. `https://github.com/fchollet/keras`, 2015.

[27] Chaeyeon Chung, Jung Soo Lee, Kyungmin Park, Junsoo Lee, Jaegul Choo, and Sungsoo Ray Hong. Understanding human-side impact of sequencing images in batch labeling for subjective tasks. *Proceedings of the ACM on Human-Computer Interaction*, (CSCW), 2021.

[28] John Joon Young Chung, Jean Y Song, Sindhu Kutty, Sungsoo Ray Hong, Juho Kim, and Walter S Lasecki. Efficient elicitation approaches to estimate collective crowd answers. *Proceedings of the ACM on Human-Computer Interaction*, 3 (CSCW):1–25, 2019.

[29] Noel CF Codella, David Gutman, M Emre Celebi, Brian Helba, Michael A Marchetti, Stephen W Dusza, Aadi Kalloo, Konstantinos Liopyris, Nabin Mishra, Harald Kittler, et al. Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic). In *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*, pages 168–172. IEEE, 2018.

[30] Dennis Collaris and Jarke J van Wijk. Explainexplore: Visual exploration of machine learning explanations. In *2020 IEEE Pacific Visualization Symposium (PacificVis)*, pages 26–35. IEEE, 2020.

[31] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

[32] Abhishek Das, Harsh Agrawal, Larry Zitnick, Devi Parikh, and Dhruv Batra. Human attention in visual question answering: Do humans and deep networks look at the same regions? *Computer Vision and Image Understanding*, 163: 90–100, 2017.

[33] Willem De Haan, Yolande AL Pijnenburg, Rob LM Strijers, Yolande van der Made, Wiesje M van der Flier, Philip Scheltens, and Cornelis J Stam. Functional neural network analysis in frontotemporal dementia and alzheimer's disease using eeg and graph theory. *BMC neuroscience*, 10(1):1–12, 2009.

[34] Russell L De Valois, E William Yund, and Norva Hepler. The orientation and direction selectivity of cells in macaque visual cortex. *Vision research*, 22(5): 531–544, 1982.

[35] Mengnan Du, Ninghao Liu, Fan Yang, and Xia Hu. Learning credible dnns via incorporating prior knowledge and model local explanation. *Knowledge and Information Systems*, 63(2):305–332, 2021.

[36] Sayna Ebrahimi, Suzanne Petryk, Akash Gokul, William Gan, Joseph E Gonzalez, Marcus Rohrbach, and Trevor Darrell. Remembering for the right reasons: Explanations reduce catastrophic forgetting. *Applied AI Letters*, 2(4):e44, 2021.

[37] Noémie Elhadad, Shaodian Zhang, Patricia Driscoll, and Samuel Brody. Characterizing the sublanguage of online breast cancer forums for medications, symptoms, and emotions. In *AMIA Annual Symposium Proceedings*, volume 2014, page 516. American Medical Informatics Association, 2014.

[38] Akiko Eriguchi, Kazuma Hashimoto, and Yoshimasa Tsuruoka. Tree-to-sequence attentional neural machine translation. *arXiv preprint arXiv:1603.06075*, 2016.

[39] Dhivya Eswaran, Christos Faloutsos, Sudipto Guha, and Nina Mishra. Spotlight: Detecting anomalies in streaming graphs. In *KDD 2018*, pages 1378–1386, 2018.

[40] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation. In *The World Wide Web Conference*, pages 417–426, 2019.

[41] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017.

[42] David J Freedman and John A Assad. Experience-dependent representation of visual categories in parietal cortex. *Nature*, 443(7107):85, 2006.

[43] Hiroshi Fukui, Tsubasa Hirakawa, Takayoshi Yamashita, and Hironobu Fujiyoshi. Attention branch network: Learning of attention mechanism for visual explanation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10705–10714, 2019.

[44] Yuyang Gao, Lingfei Wu, Houman Homayoun, and Liang Zhao. Dyngraph2seq: Dynamic-graph-to-sequence interpretable learning for health stage prediction in online health forums. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 1042–1047. IEEE, 2019.

[45] Yuyang Gao, Giorgio A Ascoli, and Liang Zhao. Bean: Interpretable and ef-

ficient learning with biologically-enhanced artificial neuronal assembly regularization. *Frontiers in Neurorobotics*, 15:68, 2021.

[46] Yuyang Gao, Tong Sun, Rishab Bhatt, Dazhou Yu, Sungsoo Hong, and Liang Zhao. Gnes: Learning to explain graph neural networks. In *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE, December 2021.

[47] Yuyang Gao, Tong Sun, Guangji Bai, Siyi Gu, Sungsoo Ray Hong, and Liang Zhao. Res: A robust framework for guiding visual explanation. In *Proceedings of the 28th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, August 2022.

[48] Yuyang Gao, Tong Sun, Liang Zhao, and Sungsoo Hong. Aligning eyes between humans and deep neural network through interactive attention alignment, 2022.

[49] Yolanda Gil, James Honaker, Shikhar Gupta, Yibo Ma, Vito D'Orazio, Daniel Garijo, Shruti Gadewar, Qifan Yang, and Neda Jahanshad. Towards human-guided machine learning. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pages 614–624, 2019.

[50] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[51] Palash Goyal, Sujit Rokka Chhetri, and Arquimedes Canedo. dyngraph2vec: Capturing network dynamics using dynamic graph representation learning. *arXiv preprint arXiv:1809.02657*, 2018.

[52] Palash Goyal, Nitin Kamra, Xinran He, and Yan Liu. Dyngem: Deep embedding method for dynamic graphs. *arXiv preprint arXiv:1805.11273*, 2018.

[53] Mark S Granovetter. The strength of weak ties. In *Social networks*, pages 347–367. Elsevier, 1977.

[54] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42, 2018.

[55] Xiaojie Guo, Lingfei Wu, and Liang Zhao. Deep graph translation. *arXiv preprint arXiv:1805.09980*, 2018.

[56] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[57] Donald O. Hebb. The organization of behavior. a neuropsychological theory. 1949.

[58] Lisa Anne Hendricks, Kaylee Burns, Kate Saenko, Trevor Darrell, and Anna Rohrbach. Women also snowboard: Overcoming bias in captioning models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 771–787, 2018.

[59] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[60] Paul W Holland and Samuel Leinhardt. Transitivity in structural models of small groups. *Comparative group studies*, 2(2):107–124, 1971.

[61] Sungsoo Ray Hong, Jessica Hullman, and Enrico Bertini. Human factors in model interpretability: Industry practices, challenges, and needs. *Proceedings of the ACM on Human-Computer Interaction*, 4:1–26, 2020.

[62] Qiang Huang, Makoto Yamada, Yuan Tian, Dinesh Singh, Dawei Yin, and Yi Chang. Graphlime: Local interpretable model explanations for graph neural networks. *arXiv preprint arXiv:2001.06216*, 2020.

[63] Aya Abdelsalam Ismail, Hector Corrada Bravo, and Soheil Feizi. Improving deep learning interpretability by saliency guided training. *Advances in Neural Information Processing Systems*, 34:26726–26739, 2021.

[64] Laurent Jacob, Guillaume Obozinski, and Jean-Philippe Vert. Group lasso with overlap and graph lasso. In *ICML 2009*, pages 433–440. ACM, 2009.

[65] Alon Jacovi and Yoav Goldberg. Aligning faithful interpretations with their social attribution. *arXiv preprint arXiv:2006.01067*, 2020.

[66] Mukund Jha and Noémie Elhadad. Cancer stage prediction based on patient online discourse. In *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing*, pages 64–71, 2010.

[67] Teja Kanchinadam, Keith Westpfahl, Qian You, and Glenn Fung. Rationale-based human-in-the-loop via supervised attention. In *DaSH@ KDD*, 2020.

[68] Akisato Kimura, Zoubin Ghahramani, Koh Takeuchi, Tomoharu Iwata, and Naonori Ueda. Few-shot learning of neural networks from scratch by pseudo example optimization. *arXiv preprint arXiv:1802.03039*, 2018.

[69] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[70] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[71] Deguang Kong, Ryohei Fujimaki, Ji Liu, Feiping Nie, and Chris Ding. Exclusive

feature learning on arbitrary structures via $\ell_{1,2}$-norm. In *Advances in Neural Information Processing Systems*, pages 1655–1663, 2014.

[72] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017.

[73] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[74] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[75] Anders Krogh and John A Hertz. A simple weight decay can improve generalization. In *Advances in neural information processing systems*, pages 950–957, 1992.

[76] David WG Langerhuizen, Anne Eva J Bulstra, Stein J Janssen, David Ring, Gino MMJ Kerkhoffs, Ruurd L Jaarsma, and Job N Doornberg. Is deep learning on par with human observers for detection of radiographically visible and occult fractures of the scaphoid? *Clinical orthopaedics and related research*, 478(11): 2653, 2020.

[77] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86 (11):2278–2324, 1998.

[78] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521 (7553):436, 2015.

[79] John Boaz Lee, Ryan Rossi, and Xiangnan Kong. Graph classification using structural attention. In *KDD 2018*, pages 1666–1674. ACM, 2018.

[80] Seungho Lee, Minhyun Lee, Jongwuk Lee, and Hyunjung Shim. Railroad is not a train: Saliency as pseudo-pixel supervision for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5495–5505, 2021.

[81] Aming Li, Sean P Cornelius, Y-Y Liu, Long Wang, and A-L Barabási. The fundamental advantages of temporal networks. *Science*, 358(6366):1042–1046, 2017.

[82] Kunpeng Li, Ziyan Wu, Kuan-Chuan Peng, Jan Ernst, and Yun Fu. Tell me where to look: Guided attention inference network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9215–9223, 2018.

[83] Yujia Li, Oriol Vinyals, Chris Dyer, Razvan Pascanu, and Peter Battaglia. Learning deep generative models of graphs. *arXiv preprint arXiv:1803.03324*, 2018.

[84] Moshe Lichman et al. Uci machine learning repository, 2013.

[85] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, 2004.

[86] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[87] Drew Linsley, Dan Shiebler, Sven Eberhardt, and Thomas Serre. Learning what and where to attend. *arXiv preprint arXiv:1805.08819*, 2018.

[88] Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. Parameterized explainer for graph neural network. *arXiv preprint arXiv:2011.04573*, 2020.

[89] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[90] Minh-Thang Luong, Eugene Brevdo, and Rui Zhao. Neural machine translation (seq2seq) tutorial. *https://github.com/tensorflow/nmt*, 2017.

[91] Yao Ma, Ziyi Guo, Zhaochun Ren, Eric Zhao, Jiliang Tang, and Dawei Yin. Dynamic graph neural networks. *arXiv preprint arXiv:1810.10627*, 2018.

[92] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[93] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[94] Matteo Mainetti and Giorgio A Ascoli. A neural mechanism for background information-gated learning based on axonal-dendritic overlaps. *PLoS computational biology*, 11(3):e1004155, 2015.

[95] Ines Filipa Martins, Ana L Teixeira, Luis Pinheiro, and Andre O Falcao. A bayesian approach to in silico blood-brain barrier penetration modeling. *Journal of chemical information and modeling*, 52(6):1686–1697, 2012.

[96] Daiki Matsunaga, Toyotaro Suzumura, and Toshihiro Takahashi. Exploring graph neural networks for stock market predictions with rolling window analysis. *arXiv preprint arXiv:1909.10660*, 2019.

[97] Andreas Mayr, Günter Klambauer, Thomas Unterthiner, and Sepp Hochreiter. Deeptox: toxicity prediction using deep learning. *Frontiers in Environmental Science*, 3:80, 2016.

[98] Masahiro Mitsuhara, Hiroshi Fukui, Yusuke Sakashita, Takanori Ogata, Tsubasa Hirakawa, Takayoshi Yamashita, and Hironobu Fujiyoshi. Embedding human knowledge into deep neural network via attention map. *arXiv preprint arXiv:1905.03540*, 2019.

[99] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern recognition*, 65:211–222, 2017.

[100] Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. Layer-wise relevance propagation: an overview. *Explainable AI: interpreting, explaining and visualizing deep learning*, pages 193–209, 2019.

[101] Ari S Morcos, David GT Barrett, Neil C Rabinowitz, and Matthew Botvinick. On the importance of single directions for generalization. *arXiv preprint arXiv:1803.06959*, 2018.

[102] Besmira Nushi, Ece Kamar, and Eric Horvitz. Towards accountable ai: Hybrid human-machine analyses for characterizing system failure. *arXiv preprint arXiv:1809.07424*, 2018.

[103] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a

method for automatic evaluation of machine translation. In *ACL 2002*, pages 311–318, 2002.

[104] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

[105] Badri Patro, Vinay Namboodiri, et al. Explanation vs attention: A two-player game to obtain attention for vqa. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11848–11855, 2020.

[106] Tejaswini Pedapati, Avinash Balakrishnan, Karthikeyan Shanmugam, and Amit Dhurandhar. Learning global transparent models consistent with local contrastive explanations. *Advances in neural information processing systems*, 33:3592–3602, 2020.

[107] Adrien Peyrache, Karim Benchenane, Mehdi Khamassi, Sidney I Wiener, and Francesco P Battaglia. Principal component analysis of ensemble recordings reveals cell assemblies at high temporal resolution. *Journal of computational neuroscience*, 29(1-2):309–325, 2010.

[108] Phillip E Pope, Soheil Kolouri, Mohammad Rostami, Charles E Martin, and Heiko Hoffmann. Explainability methods for graph convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10772–10781, 2019.

[109] Prasanna Porwal, Samiksha Pachade, Ravi Kamble, Manesh Kokare, Girish Deshmukh, Vivek Sahasrabuddhe, and Fabrice Meriaudeau. Indian diabetic retinopathy image dataset (idrid), 2018. URL `https://dx.doi.org/10.21227/H25W98`.

[110] Friedemann Pulvermüller and Andreas Knoblauch. Discrete combinatorial circuits emerging in neural networks: A mechanism for rules of grammar in the human brain? *Neural networks*, 22(2):161–172, 2009.

[111] Tingting Qiao, Jianfeng Dong, and Duanqing Xu. Exploring human-like attention supervision in visual question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[112] Colin Raffel and Daniel PW Ellis. Feed-forward networks with attention can solve some long-term memory problems. *arXiv preprint arXiv:1512.08756*, 2015.

[113] Colin Raffel, Minh-Thang Luong, Peter J Liu, Ron J Weiss, and Douglas Eck. Online and linear-time attention by enforcing monotonic alignments. *arXiv preprint arXiv:1704.00784*, 2017.

[114] Christopher L Rees, Keivan Moradi, and Giorgio A Ascoli. Weighing the evidence in peters' rule: does neuronal morphology predict connectivity? *Trends in neurosciences*, 40(2):63–71, 2017.

[115] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[116] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[117] Marta Rivera-Alba, Hanchuan Peng, Gonzalo G de Polavieja, and Dmitri B Chklovskii. Wiring economy can account for cell body placement across species and brain areas. *Current Biology*, 24(3):R109–R110, 2014.

[118] Garry Robins and Malcolm Alexander. Small worlds among interlocking directors: Network structure and distance in bipartite graphs. *Computational & Mathematical Organization Theory*, 10(1):69–94, 2004.

[119] Andrew Slavin Ross, Michael C Hughes, and Finale Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. *arXiv preprint arXiv:1703.03717*, 2017.

[120] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[121] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

[122] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[123] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in neural information processing systems*, pages 3856–3866, 2017.

[124] Farig Sadeque, Thamar Solorio, Ted Pedersen, Prasha Shrestha, and Steven Bethard. Predicting continued participation in online health forums. In *Proceedings of the Sixth International Workshop on Health Text Mining and Information Analysis*, pages 12–20, 2015.

[125] Gobinda Saha and Kaushik Roy. Saliency guided experience packing for replay in continual learning. *arXiv preprint arXiv:2109.04954*, 2021.

[126] Alexei V Samsonovich, Rebecca F Goldin, and Giorgio A Ascoli. Toward a semantic general theory of everything. *Complexity*, 15(4):12–18, 2010.

[127] Simone Scardapane, Danilo Comminiello, Amir Hussain, and Aurelio Uncini. Group sparse regularization for deep neural networks. *Neurocomputing*, 241: 81–89, 2017.

[128] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

[129] Michael Sejr Schlichtkrull, Nicola De Cao, and Ivan Titov. Interpreting graph neural networks for nlp with differentiable edge masking. *arXiv preprint arXiv:2010.00577*, 2020.

[130] Thomas Schnake, Oliver Eberle, Jonas Lederer, Shinichi Nakajima, Kristof T. Schütt, Klaus-Robert Müller, and Grégoire Montavon. Higher-order explanations of graph neural networks via relevant walks, 2020.

[131] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[132] Xiaoting Shao, Tjitze Rienstra, Matthias Thimm, and Kristian Kersting. Towards understanding and arguing with classifiers: Recent progress. *Datenbank-Spektrum*, 20(2):171–180, 2020.

[133] Haifeng Shen, Kewen Liao, Zhibin Liao, Job Doornberg, Maoying Qiao, Anton Van Den Hengel, and Johan W Verjans. Human-ai interactive and continuous sensemaking: A case study of image classification using scribble attention maps. In *Extended Abstracts of CHI*, pages 1–8, 2021.

[134] Martin Simonovsky and Nikos Komodakis. Graphvae: Towards generation of small graphs using variational autoencoders. *arXiv preprint arXiv:1802.03480*, 2018.

[135] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[136] Krishna Kumar Singh, Dhruv Mahajan, Kristen Grauman, Yong Jae Lee, Matt Feiszli, and Deepti Ghadiyaram. Don't judge an object by its context: learning to overcome contextual bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11070–11078, 2020.

[137] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017.

[138] Richard Socher, Christopher D Manning, and Andrew Y Ng. Learning continuous phrase representations and syntactic parsing with recursive neural networks. In *Proceedings of the NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop*, volume 2010, pages 1–9, 2010.

[139] Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. Zero-shot learning through cross-modal transfer. In *Advances in neural information processing systems*, pages 935–943, 2013.

[140] Ekta Sood, Simon Tannert, Philipp Müller, and Andreas Bulling. Improving natural language processing tasks with human gaze-guided neural attention. *Advances in Neural Information Processing Systems*, 33:6327–6341, 2020.

[141] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[142] Govindan Subramanian, Bharath Ramsundar, Vijay Pande, and Rajiah Aldrin Denny. Computational modeling of β-secretase 1 (bace-1) inhibitors using ligand based approaches. *Journal of chemical information and modeling*, 56(10): 1936–1949, 2016.

[143] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *NIPS 2014*, pages 3104–3112, 2014.

[144] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[145] Kai Sheng Tai, Richard Socher, and Christopher D Manning. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*, 2015.

[146] Giulio Tononi and Olaf Sporns. Measuring information integration. *BMC neuroscience*, 4(1):31, 2003.

[147] Rakshit Trivedi, Mehrdad Farajtbar, Prasenjeet Biswal, and Hongyuan Zha. Representation learning over dynamic graphs. *arXiv preprint arXiv:1803.04051*, 2018.

[148] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 1(2), 2017.

[149] Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. Order matters: Sequence to sequence for sets. *arXiv preprint arXiv:1511.06391*, 2015.

[150] Roman Visotsky, Yuval Atzmon, and Gal Chechik. Few-shot learning with per-sample rich supervision. *arXiv preprint arXiv:1906.03859*, 2019.

[151] Minh N Vu and My T Thai. Pgm-explainer: Probabilistic graphical model explanations for graph neural networks. *arXiv preprint arXiv:2010.05788*, 2020.

[152] Junxiang Wang and Liang Zhao. Multi-instance domain adaptation for vaccine adverse event detection. In *WWW 2018*, pages 97–106, 2018.

[153] Tianlu Wang, Jieyu Zhao, Mark Yatskar, Kai-Wei Chang, and Vicente Ordonez. Balanced datasets are not enough: Estimating and mitigating gender bias in deep image representations. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5310–5319, 2019.

[154] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2097–2106, 2017.

[155] Yi-Chia Wang, Robert Kraut, and John M Levine. To stay or leave?: the relationship of emotional and informational support to commitment in online health support groups. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pages 833–842.

[156] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393(6684):440, 1998.

[157] Jia Wu, Shirui Pan, Xingquan Zhu, Chengqi Zhang, and S Yu Philip. Multiple structure-view learning for graph classification. *IEEE Transactions on Neural Networks and Learning Systems*, 29(7):3236–3251, 2018.

[158] Lingfei Wu, Ian En-Hsu Yen, Zhen Zhang, Kun Xu, Liang Zhao, Xi Peng, Yinglong Xia, and Charu Aggarwal. Scalable global alignment graph kernel using random features: From node embedding to graph embedding. In *KDD 2019*, pages 1418–1428, 2019.

[159] Lingfei Wu, Peng Cui, Jian Pei, and Liang Zhao. *Graph Neural Networks: Foundations, Frontiers, and Applications.* Springer, Singapore, 2021.

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

[161] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[162] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[163] Kun Xu, Lingfei Wu, Zhiguo Wang, Yansong Feng, and Vadim Sheinin. Sql-to-text generation with graph-to-sequence model. *arXiv preprint arXiv:1809.05255*, 2018.

[164] Kun Xu, Lingfei Wu, Zhiguo Wang, and Vadim Sheinin. Graph2seq: Graph to sequence learning with attention-based neural networks. *arXiv preprint arXiv:1804.00823*, 2018.

[165] Kun Xu, Lingfei Wu, Zhiguo Wang, Mo Yu, Liwei Chen, and Vadim Sheinin. Exploiting rich syntactic information for semantic parsing with graph-to-sequence model. *arXiv preprint arXiv:1808.07624*, 2018.

[166] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. *arXiv preprint arXiv:1808.00191*, 2, 2018.

[167] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *NAACL*, pages 1480–1489, 2016.

[168] Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems*, 32:9240, 2019.

[169] Jaehong Yoon and Sung Ju Hwang. Combined group and exclusive sparsity for deep neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3958–3966. JMLR. org, 2017.

[170] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing. *ieee Computational intelligenCe magazine*, 13(3):55–75, 2018.

[171] Fuxun Yu, Zhuwei Qin, Chenchen Liu, Liang Zhao, Yanzhi Wang, and Xiang Chen. Interpreting and evaluating neural network robustness. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 4199–4205, 2019.

[172] Wenchao Yu, Wei Cheng, Charu C Aggarwal, Kai Zhang, Haifeng Chen, and Wei Wang. Netwalk: A flexible deep embedding approach for anomaly detection in dynamic networks. In *KDD 2018*, pages 2672–2681. ACM, 2018.

[173] Hao Yuan, Jiliang Tang, Xia Hu, and Shuiwang Ji. Xgnn: Towards model-level explanations of graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 430–438, 2020.

[174] Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji. Explainability in graph neural networks: A taxonomic survey. *arXiv preprint arXiv:2012.15445*, 2020.

[175] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[176] Omar Zaidan, Jason Eisner, and Christine Piatko. Using "annotator rationales" to improve machine learning for text categorization. In *Human language technologies 2007: The conference of the North American chapter of the association for computational linguistics; proceedings of the main conference*, pages 260–267, 2007.

[177] Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126(10):1084–1102, 2018.

[178] Quan-shi Zhang and Song-Chun Zhu. Visual interpretability for deep learning: a survey. *Frontiers of Information Technology & Electronic Engineering*, 19(1): 27–39, 2018.

[179] Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. Interpretable convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8827–8836, 2018.

[180] Shaodian Zhang and Noémie Elhadad. Factors contributing to dropping-out in an online health community: Static and longitudinal analyses. In *AMIA Annual Symposium Proceedings*, volume 2016, page 2090. American Medical Informatics Association, 2016.

[181] Shaodian Zhang, Erin Bantum, Jason Owen, and Noémie Elhadad. Does sus-

tained participation in an online health community affect sentiment? In *AMIA Annual Symposium Proceedings*, volume 2014, page 1970, 2014.

[182] Shaodian Zhang, Edouard Grave, Elizabeth Sklar, and Noemie Elhadad. Longitudinal analysis of discussion topics in an online breast cancer community using convolutional neural networks. *Journal of biomedical informatics*, 69:1–9, 2017.

[183] Yue Zhang, David Defazio, and Arti Ramesh. Relex: A model-agnostic relational model explainer. *arXiv preprint arXiv:2006.00305*, 2020.

[184] Yundong Zhang, Juan Carlos Niebles, and Alvaro Soto. Interpretable visual question answering by visual grounding from attention supervision mining. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 349–357. IEEE, 2019.

[185] Ziwei Zhang, Peng Cui, Jian Pei, Xiao Wang, and Wenwu Zhu. Timers: Error-bounded svd restart on dynamic networks. *arXiv preprint arXiv:1711.09541*, 2017.

[186] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. *arXiv preprint arXiv:1707.09457*, 2017.

[187] Liang Zhao, Jieping Ye, Feng Chen, Chang-Tien Lu, and Naren Ramakrishnan. Hierarchical incomplete multi-source feature learning for spatiotemporal event forecasting. In *KDD 2016*, pages 2085–2094. ACM, 2016.

[188] Zejia Zheng and Juyang Weng. Challenges in visual parking and how a developmental network approaches the problem. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 4593–4600. IEEE, 2016.

[189] Zejia Zheng and Juyang Weng. Mobile device based outdoor navigation with on-line learning neural network: A comparison with convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 11–18, 2016.

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

[191] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *TPAMI*, 2017.

[192] Bolei Zhou, Yiyou Sun, David Bau, and Antonio Torralba. Revisiting the importance of individual units in cnns via ablation. *arXiv preprint arXiv:1806.02891*, 2018.

[193] Le-kui Zhou, Yang Yang, Xiang Ren, Fei Wu, and Yueting Zhuang. Dynamic network embedding by modeling triadic closure process. In *AAAI*, 2018.

[194] Yang Zhou, Rong Jin, and Steven Chu-Hong Hoi. Exclusive lasso for multi-task feature selection. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 988–995, 2010.

[195] Linhong Zhu, Dong Guo, Junming Yin, Greg Ver Steeg, and Aram Galstyan. Scalable temporal latent space inference for link prediction in dynamic social networks. *TKDE*, 28(10):2765–2777, 2016.

[196] Jiaxin Zhuang, Jiabin Cai, Ruixuan Wang, Jianguo Zhang, and Weishi Zheng. Care: Class attention to regions of lesion for classification on imbalanced data.

In *International Conference on Medical Imaging with Deep Learning*, pages 588–597. PMLR, 2019.