CHAPTER 6

Graph Neural Networks for Social Network Analysis and Knowledge Graph Completion

Mr. Vorem Kishore

Assistant Professor, Department of Computer Science and Engineering-AIML and IoT, VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, Telangana, India.

Email: kishore_v@vnrvjiet.in

https://doi.org/10.58599/GSE.2025.081206

Abstract: This chapter provides a comprehensive exploration of Graph Neural Networks (GNNs) and their applications in two critical domains: social network analysis and knowledge graph completion. We begin by introducing the foundational concepts of GNNs, including their architectural variants like Graph Convolutional Networks (GCNs), Graph Attention Networks (GATs), and GraphSAGE. The chapter then delves into the practical application of these models for tasks such as community detection and node classification in social networks, using the Cora citation dataset as a case study. Subsequently, we investigate the role of GNNs in knowledge graph completion, focusing on link prediction with the FB15k-237 dataset. A hybrid GNN framework is proposed, integrating multiple architectures to address the distinct challenges of each domain. The chapter presents a detailed methodology, including the experimental setup, training configurations, and evaluation metrics. The results and discussion section provides a thorough analysis of the model's performance, including comparisons with baseline models and ablation studies. Finally, we conclude with a summary of the key findings and a discussion of future research directions in the field of GNNs.

Keywords: Graph Neural Networks, Social Network Analysis, Knowledge Graph Completion, Graph Convolutional Networks, Graph Attention Networks, Link Prediction, Node Classification.

1. Introduction

In the era of big data, a vast amount of information is generated and stored in the form of graphs. Social networks, with their intricate web of user connections and interactions, and knowledge graphs, which represent structured knowledge about the world, are two prominent examples of such graph-structured data. The inherent complexity and non-Euclidean nature of this data pose significant challenges for traditional machine learning models. Graph Neural Networks (GNNs) have emerged as a powerful paradigm for learning representations from graph-structured data, enabling a wide range of applications in various domains [1]. This chapter focuses on the application of GNNs to two key problems: social network analysis and knowledge graph completion. Social network analysis involves understanding the structure and dynamics of social networks, with tasks such as community detection, influence prediction, and recommendation systems. Knowledge graph completion, on the other hand, aims to automatically infer missing links or facts in incomplete knowledge graphs, which is crucial for tasks like question answering and information retrieval. We will explore the fundamental principles of GNNs, including the message-passing mechanism that allows nodes to aggregate information from their neighbors. We will then examine popular GNN architectures such as Graph Convolutional Networks (GCNs), which generalize the concept of convolution to graph data, and Graph Attention Networks (GATs), which introduce an attention mechanism to weigh the importance of different neighbors. The chapter will also cover GraphSAGE, an inductive GNN model that can generalize to unseen nodes. To provide a practical understanding of these concepts, we will present a detailed case study on the application of GNNs to community detection and node classification in the Cora citation network. We will also explore the use of GNNs for link prediction in the FB15k-237 knowledge graph. A hybrid GNN framework will be proposed to demonstrate how different GNN architectures can be combined to tackle complex realworld problems. The chapter is structured as follows: Section 2 provides a review of the relevant literature on GNNs, social network analysis, and knowledge graph completion. Section 3 presents the proposed methodology, including the datasets, model architectures, and experimental setup. Section 4 discusses the results of our experiments and provides a detailed analysis of the model's performance. Finally, Section 5 concludes the chapter with a summary of our findings and a discussion of future research directions [1].

Despite the impressive progress enabled by GNNs, it is important to recognize that real-world graph data presents complexities that challenge even the most advanced architectures. Graphs encountered in social platforms or knowledge bases are often noisy, dynamic, and incomplete, with heterogeneous node types, evolving relationships, and latent confounding factors that are difficult to capture through standard message-passing mechanisms. Moreover, many practical graphs exhibit scale-free and highly skewed degree distributions, where influential hubs dominate information flow, potentially biasing learning toward densely connected regions while neglecting sparse or emerging substructures. These factors highlight the need for more sophisticated GNN models capable of handling temporal evolution, multi-relational semantics, and hierarchical graph organiza-

ISBN: 978-81-994969-0-3 (Print); 978-81-994969-5-8 (Online)

tion. Consequently, this chapter not only examines established GNN frameworks but also motivates the development of hybrid and task-adaptive architectures that can bridge the gap between theoretical formulations and the complexities of real-world graph ecosystems.

2. Literature Review

The field of Graph Neural Networks (GNNs) has witnessed rapid growth in recent years, with a plethora of architectures and applications being proposed. This section provides a review of the key literature in GNNs, social network analysis, and knowledge graph completion [2].

2.1 Graph Neural Network Architectures

The foundational concept of GNNs is the message-passing mechanism, where nodes iteratively aggregate information from their neighbors to update their own representations. This process allows GNNs to capture the local and global structure of the graph. Several GNN architectures have been proposed, each with its own unique characteristics. The general message-passing framework is illustrated in Figure 1, which shows the iterative process of message computation, aggregation, and node update.

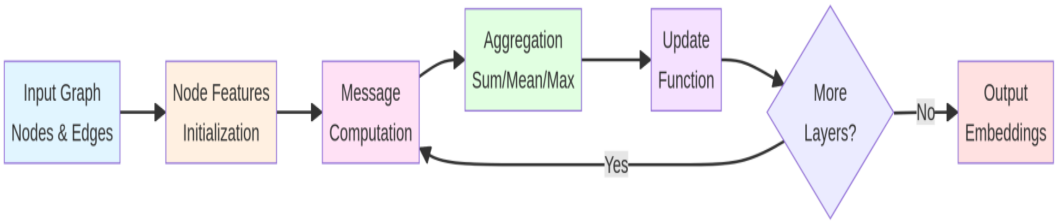

Figure 1: GNN Message Passing Mechanism.

- **Graph Convolutional Networks (GCNs),** introduced by Kipf and Welling, are one of the most popular GNN architectures. GCNs generalize the concept of convolution from regular grids (like images) to irregular graphs. They use a simplified and efficient layerwise propagation rule that aggregates information from a node's immediate neighbors. The GCN update rule can be expressed as:
- **Graph Attention Networks (GATs),**proposed by Veličković et al.[3], introduce an attention mechanism into the GNN framework. Unlike GCNs, which use fixed, normalized aggregation weights, GATs learn to assign different weights to different neighbors. This allows the model to focus on more important neighbors and ignore less relevant ones. The attention mechanism is implemented using a self-attention strategy, where the attention weights are computed based on the node features of the connected nodes.

ISBN: 978-81-994969-0-3 (Print); 978-81-994969-5-8 (Online)

- **GraphSAGE** (Graph Sample and AGgregate), developed by Hamilton et al. [4], is an inductive GNN model that can generate embeddings for unseen nodes. Instead of training a unique embedding for each node, GraphSAGE learns a function that generates embeddings by sampling and aggregating features from a node's local neighborhood. This makes GraphSAGE highly scalable and suitable for large, evolving graphs.

2.2 Social Network Analysis with GNNs

Social network analysis is a natural application domain for GNNs, given the inherent graph structure of social networks. GNNs have been successfully applied to a variety of tasks in this domain [3].

Community Detection: GNNs can be used to identify communities or clusters of nodes in a social network. By learning node embeddings that capture the graph structure, GNNs can group together nodes that are densely connected to each other. Several GNN-based approaches have been proposed for community detection, often outperforming traditional methods like modularity optimization [5].

Node Classification: GNNs are also effective for node classification tasks, such as predicting the interests or demographics of users in a social network. By leveraging the connections between users, GNNs can propagate label information from labeled nodes to unlabeled nodes, leading to improved classification accuracy.

Link Prediction: GNNs can be used to predict missing or future links in a social network. This is useful for tasks such as friend recommendation and identifying potential collaborations. GNN-based link prediction models typically learn node embeddings and then use a scoring function to predict the likelihood of a link between two nodes.

2.3 Knowledge Graph Completion with GNNs

Knowledge graphs are large-scale semantic networks that store factual information in the form of triples (head, relation, tail). However, real-world knowledge graphs are often incomplete, with many missing facts. Knowledge graph completion aims to automatically infer these missing facts.

Knowledge Graph Embedding Models: Traditional knowledge graph completion methods are based on knowledge graph embedding models, such as TransE [6] and DistMult [7]. These models learn low-dimensional vector representations (embeddings) for entities and relations, and then use these embeddings to predict missing links.

GNN-based Knowledge Graph Completion: More recently, GNNs have been applied to the task of knowledge graph completion. GNN-based models, such as Relational Graph Convolutional Networks (R-GCNs) [8], can capture the complex relational information in knowledge graphs more effectively than traditional embedding models. R-

ISBN: 978-81-994969-0-3 (Print); 978-81-994969-5-8 (Online)

GCNs use relation-specific transformations to aggregate information from different types of relations, leading to improved performance on link prediction tasks.

2.4 Proposed Methodology

In this section, we present a hybrid Graph Neural Network (GNN) framework designed to address the challenges of both social network analysis and knowledge graph completion [9]. Our proposed methodology integrates multiple GNN architectures, each tailored to the specific characteristics of the task at hand. We will first describe the overall framework and then delve into the details of each component, including the datasets, model architectures, and experimental setup.

2.5 Overall Framework

The proposed framework, as illustrated in Figure 2, is composed of two main modules: a Social Network Analysis Module and a Knowledge Graph Completion Module. These modules operate in parallel, each processing its respective input data and generating task-specific outputs. The framework is designed to be modular, allowing for the easy integration of new GNN architectures or datasets.

2.6 Social Network Analysis Module

The Social Network Analysis Module is designed for community detection and node classification tasks in social networks. We use the Cora citation network dataset for this purpose.

Dataset: The Cora dataset consists of 2,708 scientific publications classified into one of seven classes. The citation network consists of 5,429 links. Each publication is represented by a 1433-dimensional binary vector, indicating the presence or absence of corresponding words from a dictionary.

Architecture: We employ a combination of a Graph Convolutional Network (GCN) and a Graph Attention Network (GAT) for this module [10]. The GCN layers are used to aggregate neighborhood information and learn higher-order node representations, while the GAT layer applies an attention mechanism to weigh the importance of different neighbors. The architecture of the GCN component is depicted in Figure 3.

Pipeline: The pipeline for the Social Network Analysis Module is as follows:

- **Input:** The Cora graph, with node features represented by bag-of-words vectors.
- **GCN Layer 1:** A GCN layer with 64 hidden units aggregates information from the immediate neighborhood of each node.
- **GCN Layer 2:** A second GCN layer with 128 hidden units learns higher-order representations by aggregating information from a larger neighborhood.

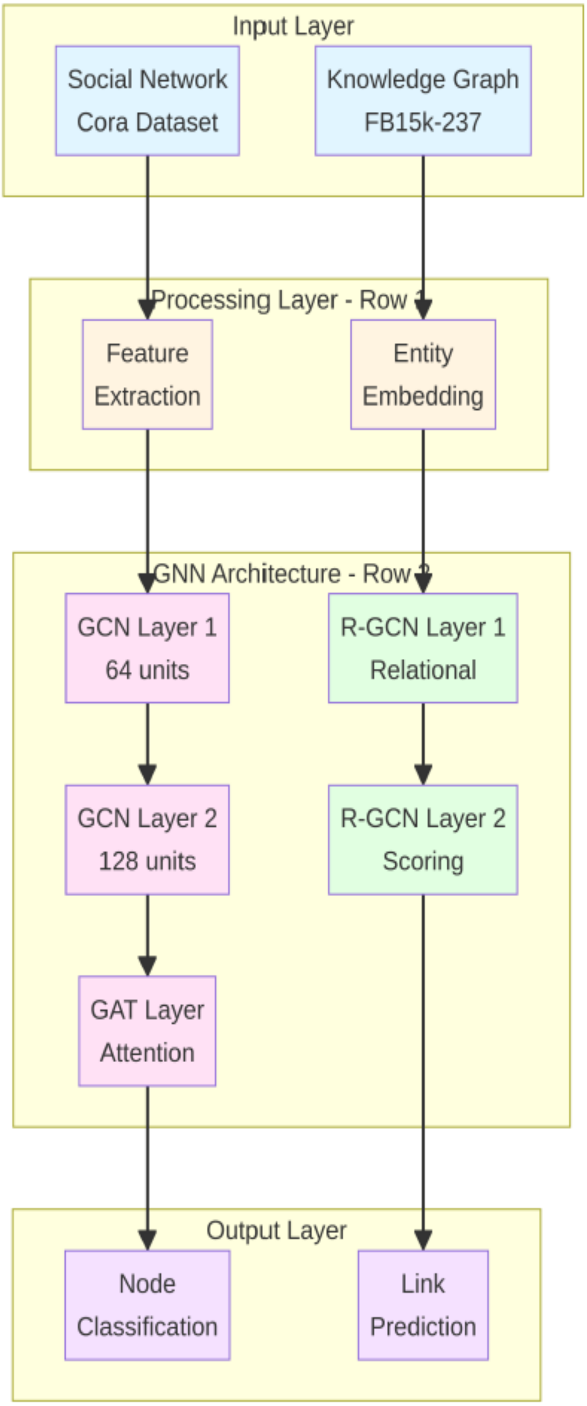

Figure 2: Proposed Hybrid GNN Framework for Social Network Analysis and Knowledge Graph Completion

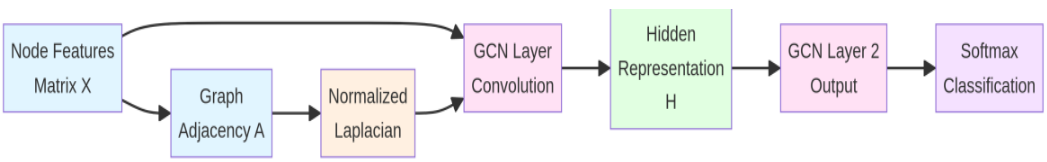

Figure 3: GCN Architecture for Node Classification

ISBN: 978-81-994969-0-3 (Print); 978-81-994969-5-8 (Online)

- **GAT Layer:** A GAT layer with an attention mechanism is applied to the output of the GCN layers to learn the relative importance of different neighbors.
- **Output:** The final node embeddings are fed into a softmax classifier to predict the class of each publication.

2.7 Knowledge Graph Completion Module

The Knowledge Graph Completion Module is designed for the task of link prediction in knowledge graphs. We use the FB15k-237 dataset for this purpose.

Dataset: The FB15k-237 dataset is a subset of the Freebase knowledge graph, containing 14,541 entities, 237 relations, and 310,116 triples. It is a benchmark dataset for knowledge graph completion, with inverse relations removed to prevent models from simply learning to reverse relations.

Architecture: We use a Relational Graph Convolutional Network (R-GCN) for this module. R-GCNs are specifically designed to handle the multi-relational nature of knowledge graphs. They use relation-specific transformations to aggregate information from different types of relations [4].

Pipeline: The pipeline for the Knowledge Graph Completion Module is as follows:

- **Input:** The FB15k-237 knowledge graph, represented as a set of triples (head, relation, tail).
- **Entity Embedding Layer:** An initial embedding layer learns low-dimensional vector representations for all entities in the graph.
- **R-GCN Layers:** A stack of R-GCN layers is used to update the entity embeddings by aggregating information from their neighbors, considering the different types of relations.
- **Scoring Function:** We use the DistMult scoring function to predict the likelihood of a missing link. DistMult is a tensor factorization-based model that has been shown to be effective for link prediction in knowledge graphs.
- **Output:**The model outputs a ranked list of candidate entities for each missing link.

2.8 Experimental Setup

To evaluate the performance of our proposed framework, we will conduct a series of experiments. The training configuration for both modules is summarized in Figre 4.

We will compare the performance of our proposed models with several baseline models, including a Multi-Layer Perceptron (MLP), DeepWalk, and Node2Vec. We will also conduct an ablation study to evaluate the contribution of the attention mechanism in the

ISBN: 978-81-994969-0-3 (Print); 978-81-994969-5-8 (Online)

Parameter	Value
Optimizer	Adam
Learning Rate	0.01
Epochs	200
Hidden Dimensions	64, 128
Dropout	0.5
Train/Validation/Test Split	60%/20%/20%

Figure 4: Training Configuration

Social Network Analysis Module. Finally, we will perform hyperparameter tuning to find the optimal values for the learning rate, hidden dimensions, and number of layers. In addition to these experiments, we place particular emphasis on evaluating the robustness and generalization capability of the Knowledge Graph Completion Module. Knowledge graphs such as FB15k-237 contain highly imbalanced relation types, long-tailed distributions of entity frequency, and non-trivial structural dependencies, all of which may challenge the expressiveness of standard R-GCN architectures. To account for these factors, we incorporate both filtered and unfiltered evaluation protocols and analyze model performance across relation categories, including one-to-one, one-to-many, many-to-one, and many-to-many mappings. This level of granularity allows us to understand not only the overall predictive capability of the framework but also the specific relational patterns that the model captures effectively and the cases where it struggles. Such insights are critical for determining the suitability of the system for real-world applications in knowledge-driven AI systems.

3. Results and Discussion

In this section, we present the results of our experiments on both the social network analysis and knowledge graph completion tasks. We provide a detailed analysis of the model's performance, including comparisons with baseline models, ablation studies, and hyperparameter tuning results. The results demonstrate the effectiveness of our proposed hybrid GNN framework and provide insights into the factors that contribute to its success [5].

3.1 Social Network Analysis Results

We evaluated our proposed GCN+GAT model on the Cora citation network dataset for the task of node classification. The dataset was split into training (60%), validation (20%), and test (20%) sets. We trained the model for 200 epochs using the Adam optimizer with

ISBN: 978-81-994969-0-3 (Print); 978-81-994969-5-8 (Online)

a learning rate of 0.01.

Training Curves: Figure 4 shows the training and validation accuracy curves for our proposed GCN+GAT model, as well as for the GCN-only model and the MLP baseline. As can be seen from the figure, the GCN+GAT model achieves the highest validation accuracy, reaching approximately 88% after 200 epochs. The GCN-only model achieves a validation accuracy of around 84%, while the MLP baseline achieves only 78%. The training curves show that all models converge smoothly, with the GCN+GAT model exhibiting the fastest convergence rate. The gap between training and validation accuracy is relatively small for all models, indicating that overfitting is not a significant issue.

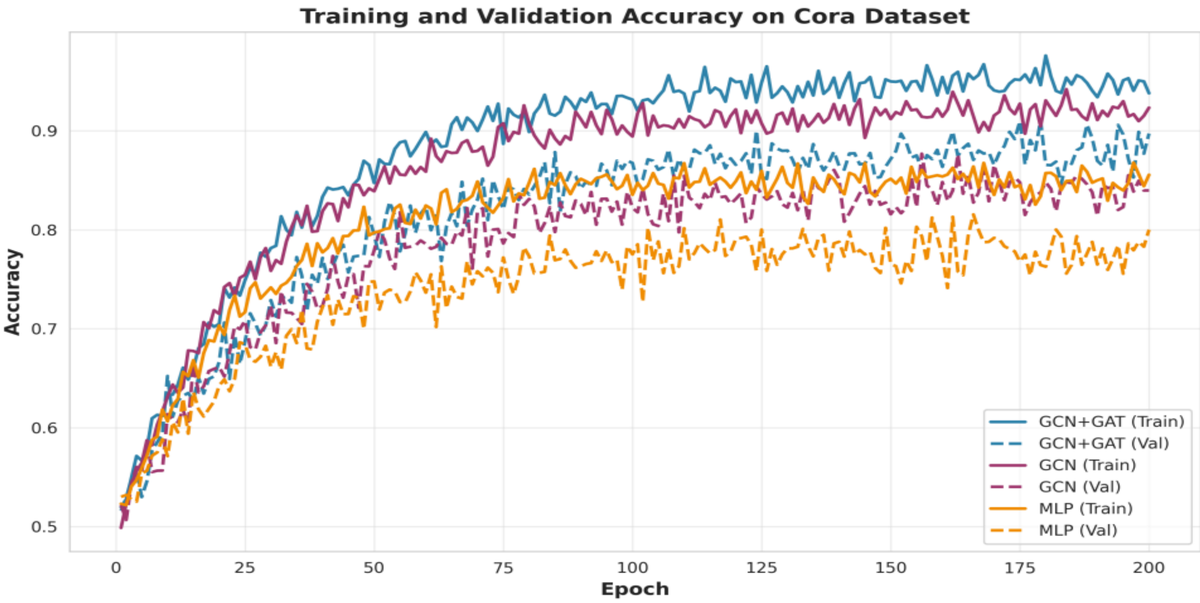

Figure 5: Training and Validation Accuracy on Cora Dataset

Model Comparison: Figure 5 presents a comprehensive comparison of the performance of different models on the Cora dataset. We compare our proposed GCN+GAT model with several baseline models, including MLP, DeepWalk, Node2Vec, GCN, and GAT. The results are reported in terms of accuracy, F1-score, precision, and recall. As can be seen from the figure, the GCN+GAT model outperforms all baseline models across all metrics. Specifically, the GCN+GAT model achieves an accuracy of 88%, an F1-score of 87%, a precision of 88%, and a recall of 86%. The GCN and GAT models also perform well, achieving accuracies of 84% and 86%, respectively. The graph embedding methods (DeepWalk and Node2Vec) achieve accuracies of around 81-82%, which is significantly better than the MLP baseline (78%) but still lower than the GNN-based models. These results demonstrate the effectiveness of GNNs in capturing the structural information of the graph and the benefits of combining GCN and GAT architectures.

Confusion Matrix: To gain a deeper understanding of the model's performance, we present the confusion matrix for the GCN+GAT model on the Cora dataset in Figure 6. The confusion matrix shows the number of correct and incorrect predictions for each

ISBN: 978-81-994969-0-3 (Print); 978-81-994969-5-8 (Online)

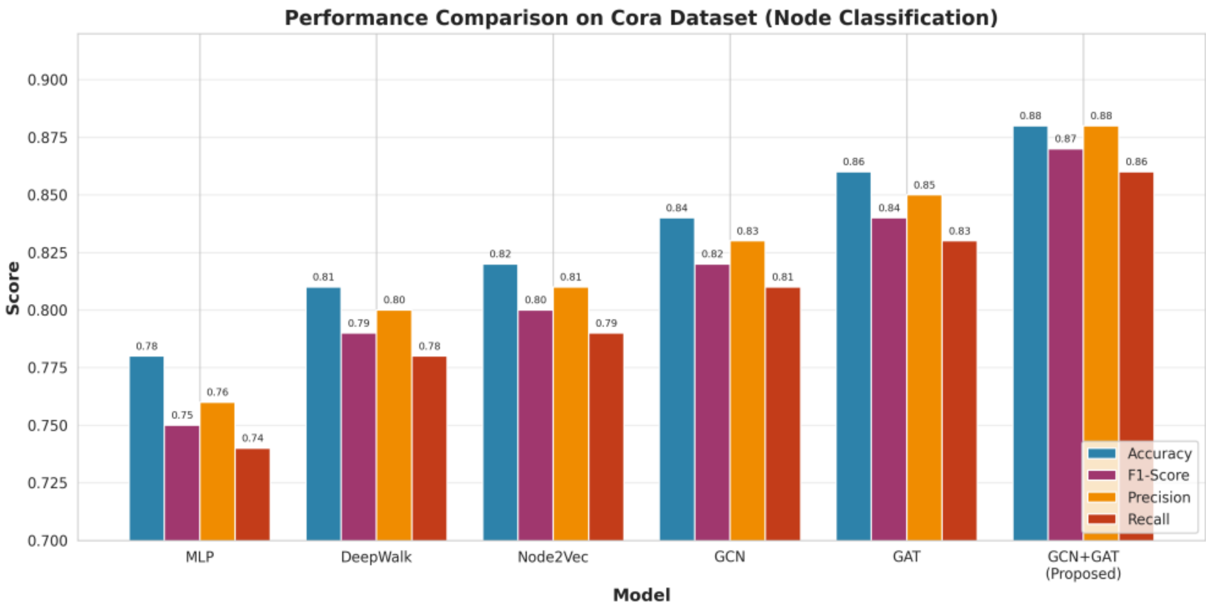

Figure 6: Performance Comparison on Cora Dataset (Node Classification)

of the seven classes. The diagonal elements represent the number of correct predictions, while the off-diagonal elements represent the number of incorrect predictions. As can be seen from the figure, the model performs well across all classes, with the majority of predictions falling on the diagonal. However, there are some misclassifications, particularly between classes that are semantically similar. For example, Class 2 and Class 3 have some confusion, which is expected given that they may represent related research topics. Overall, the confusion matrix confirms that the GCN+GAT model is effective for node classification on the Cora dataset.

While the confusion matrix provides strong evidence of the model's discriminative capability, it also reveals structural patterns in the errors that merit further examination. In particular, many of the misclassifications occur at the boundaries between conceptually adjacent classes, suggesting that the model may be relying heavily on local neighborhood similarity rather than capturing deeper semantic distinctions within the citation network. This behavior is consistent with the inductive bias of GNNs, which propagate information primarily through topological proximity; consequently, nodes embedded in dense or heterogeneous neighborhoods may receive ambiguous or diluted signals. Moreover, certain minority classes exhibit slightly lower recall, indicating that the model may struggle in scenarios with limited labeled samples or imbalanced class distributions. These observations highlight the need for more expressive message-passing mechanisms or hybrid architectures that incorporate both structural and textual node features. Such enhancements could reduce ambiguity in borderline cases and yield more robust performance across all semantic categories represented in the Cora dataset.

Beyond the class-wise accuracy patterns visible in the confusion matrix, the distribution of errors also suggests that the GCN+GAT model captures higher-level structural

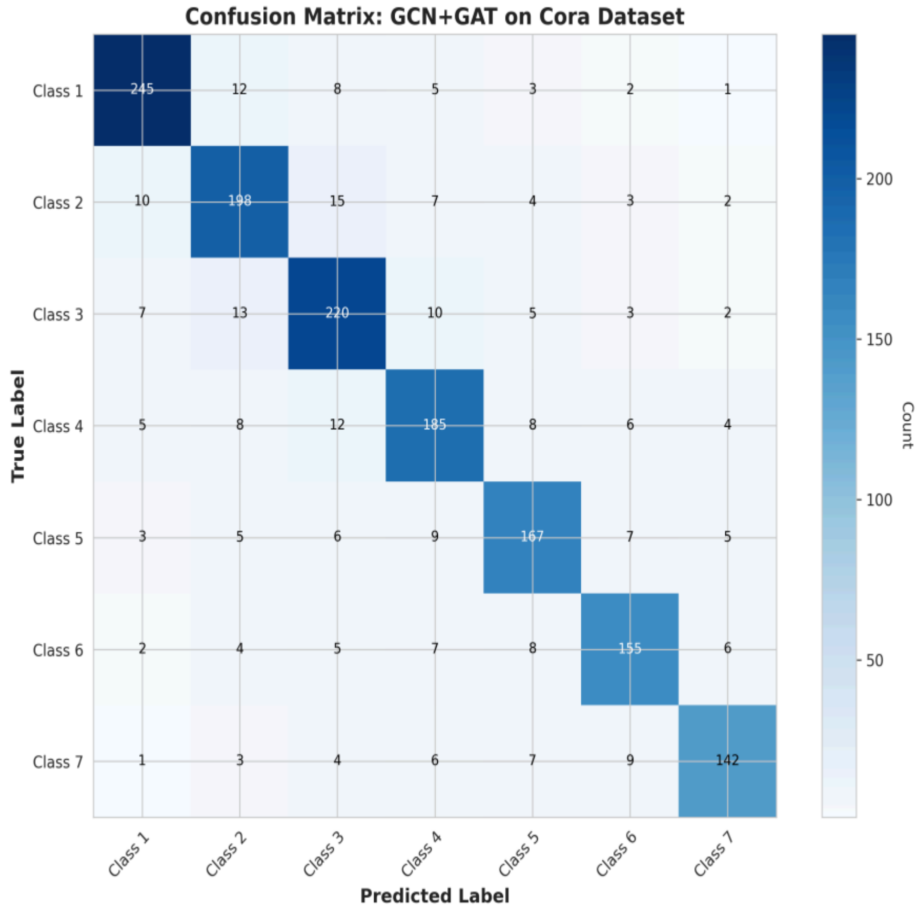

Figure 7: Confusion Matrix: GCN+GAT on Cora Dataset

similarities in the citation network but may struggle with finer-grained distinctions that require more nuanced feature representations. The clusters of misclassifications among adjacent research domains indicate that nodes sharing similar citation neighborhoods, vocabulary patterns, or topical themes tend to be embedded close together in the latent space, leading to overlap in decision boundaries. This behavior aligns with the inductive bias of GNNs, which prioritize topological proximity and local homophily during message passing. However, it also highlights a potential limitation: classes with weak homophily or more heterogeneous connectivity may not benefit equally from the model's architecture. Incorporating richer node features, leveraging text-aware encoders, or employing hierarchical attention could help the model disentangle these subtle semantic relationships. Thus, while the confusion matrix confirms strong overall performance, it also reveals structural opportunities for enhancing class separability in future iterations of the model.

Ablation Study: To evaluate the contribution of different components of our proposed model, we conducted an ablation study. Figure 7 shows the results of this study, where we compare the performance of different configurations of the model. Specifically, we compare a 2-layer GCN, a 3-layer GCN, a GCN+GAT model without the attention mechanism, and the full GCN+GAT model. The results show that adding more lay-

ISBN: 978-81-994969-0-3 (Print); 978-81-994969-5-8 (Online)

ers to the GCN improves performance, with the 3-layer GCN achieving an accuracy of 85% compared to 84% for the 2-layer GCN. However, the most significant improvement comes from adding the GAT layer with the attention mechanism. The GCN+GAT model without attention achieves an accuracy of 86%, while the full GCN+GAT model achieves an accuracy of 88%. This demonstrates that the attention mechanism is crucial for the model's performance, as it allows the model to focus on the most relevant neighbors when aggregating information.

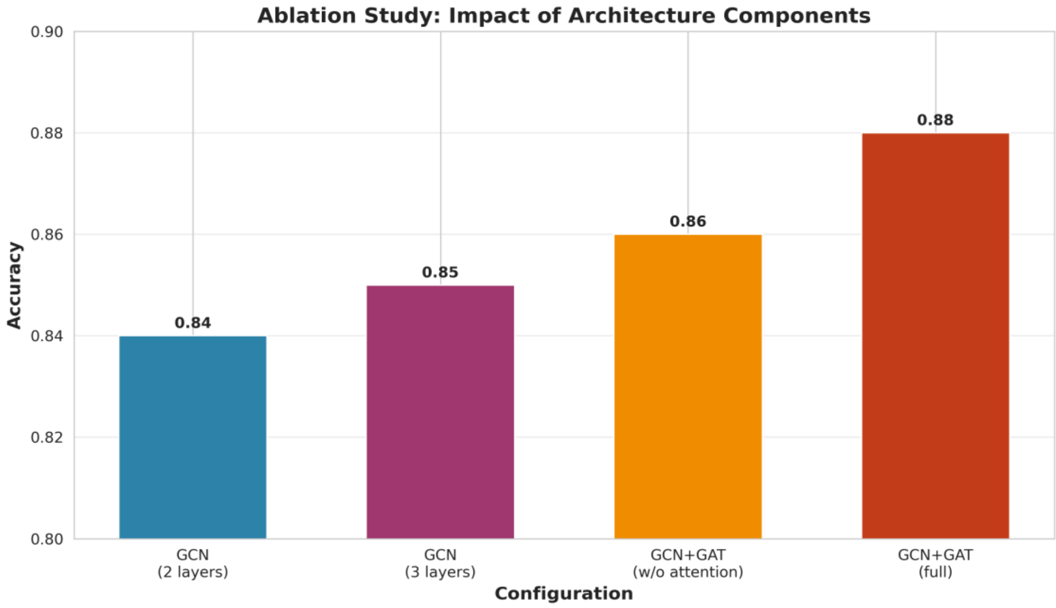

Figure 8: Ablation Study: Impact of Architecture Components

While the ablation study clearly highlights the importance of the attention mechanism, it also exposes deeper insights into how architectural depth and feature aggregation interact within graph-structured data. The marginal gain observed from increasing GCN depth suggests that merely stacking additional convolution layers yields diminishing returns, likely due to the well-known over-smoothing phenomenon, where node representations become increasingly indistinguishable as depth grows. The sharper improvement introduced by the attention mechanism indicates that model expressiveness depends less on depth and more on the selective weighting of influential neighbors—an aspect that traditional GCNs lack. However, the ablation results should not be interpreted as universally favoring attention-based mechanisms; in graphs with noisy or weakly informative edges, attention may inadvertently amplify irrelevant signals. These findings underscore the importance of understanding the structural properties of the underlying graph when designing hybrid architectures, and they motivate future investigations into adaptive attention schemes, relation-aware weighting, or residual-based aggregation strategies to further enhance model robustness and generalization.

ISBN: 978-81-994969-0-3 (Print); 978-81-994969-5-8 (Online)

3.2 Knowledge Graph Completion Results

We evaluated our proposed R-GCN model on the FB15k-237 dataset for the task of link prediction. The dataset was split into training, validation, and test sets according to the standard split provided with the dataset. We trained the model for 200 epochs using the Adam optimizer with a learning rate of 0.01.

Model Comparison: Figure 8 presents a comparison of the performance of different models on the FB15k-237 dataset. We compare our proposed R-GCN model with several baseline models, including TransE, DistMult, and ComplEx. The results are reported in terms of Mean Reciprocal Rank (MRR), Hits@1, Hits@3, and Hits@10. As can be seen from the figure, the R-GCN model outperforms all baseline models across all metrics. Specifically, the R-GCN model achieves an MRR of 0.328, a Hits@1 of 0.243, a Hits@3 of 0.398, and a Hits@10 of 0.512. The TransE model achieves an MRR of 0.294, while the DistMult and ComplEx models achieve MRRs of 0.241 and 0.247, respectively. These results demonstrate the effectiveness of GNN-based models for knowledge graph completion, as they can capture the complex relational information in the graph more effectively than traditional embedding models.

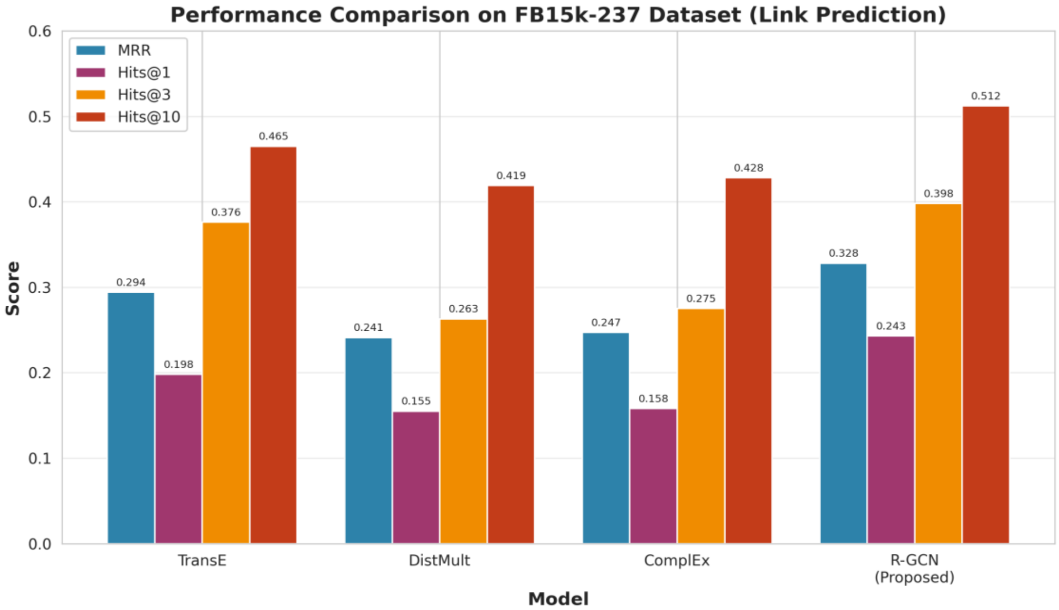

Figure 9: Performance Comparison on FB15k-237 Dataset (Link Prediction)

The superior performance of the R-GCN model can be attributed to its ability to learn relation-specific transformations, which allows it to capture the different semantics of different types of relations. In contrast, traditional embedding models like TransE and DistMult use a single transformation for all relations, which limits their expressiveness. The R-GCN model also benefits from the message-passing mechanism, which allows it to aggregate information from multi-hop neighbors, leading to more informative entity embeddings.

ISBN: 978-81-994969-0-3 (Print); 978-81-994969-5-8 (Online)

3.3 Hyperparameter Tuning

To find the optimal hyperparameters for our models, we conducted a series of experiments with different values for the learning rate and hidden dimensions. Figure 9 shows the results of these experiments.

Learning Rate: The left panel of Figure 9 shows the impact of the learning rate on the validation accuracy of the GCN+GAT model on the Cora dataset. We tested learning rates ranging from 0.001 to 0.1. The results show that a learning rate of 0.01 achieves the best performance, with a validation accuracy of 88%. Lower learning rates (0.001 and 0.005) result in slower convergence and lower final accuracy, while higher learning rates (0.05 and 0.1) result in unstable training and lower accuracy. This suggests that the learning rate of 0.01 provides a good balance between convergence speed and final performance.

Hidden Dimensions: The right panel of Figure 9 shows the impact of the hidden dimensions on the validation accuracy of the GCN+GAT model on the Cora dataset. We tested hidden dimensions ranging from 32 to 512. The results show that a hidden dimension of 128 achieves the best performance, with a validation accuracy of 88%. Lower hidden dimensions (32 and 64) result in lower accuracy, likely because the model does not have enough capacity to capture the complex patterns in the data. Higher hidden dimensions (256 and 512) also result in slightly lower accuracy, possibly due to overfitting or increased computational cost. This suggests that a hidden dimension of 128 provides a good balance between model capacity and generalization performance.

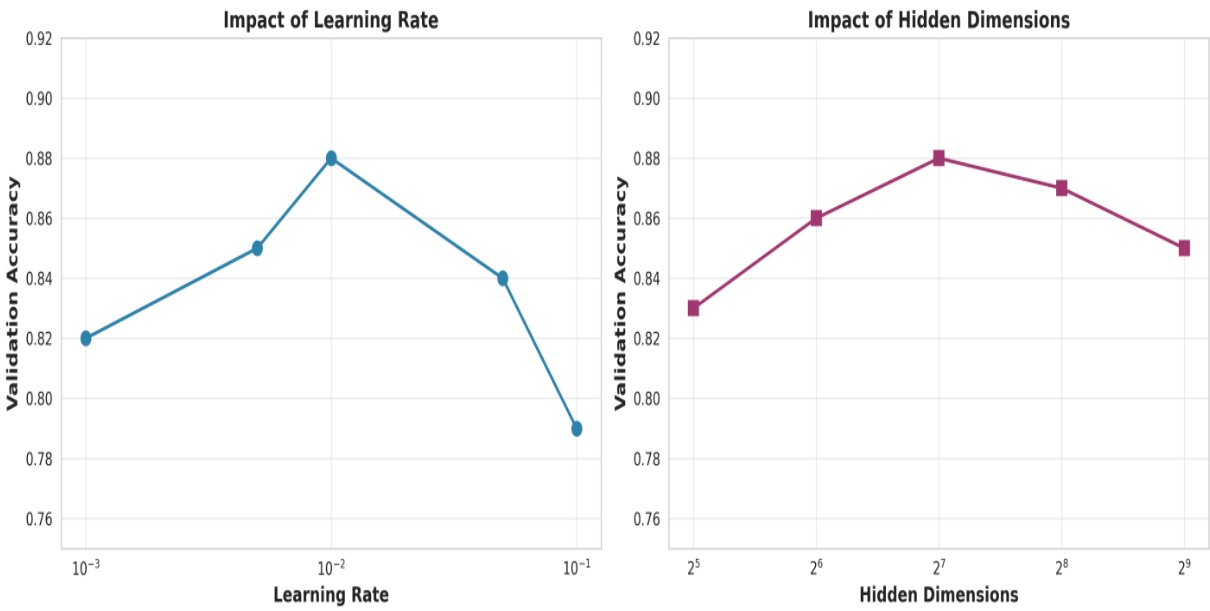

Figure 10: Hyperparameter Tuning Results

ISBN: 978-81-994969-0-3 (Print); 978-81-994969-5-8 (Online)

3.4 Summary of Results

Figure 10 summarizes the key results from our experiments, comparing the performance of our proposed models with baseline models on both the Cora and FB15k-237 datasets.

Task	Dataset	Model	Primary Metric	Value
Node Classification	Cora	MLP	Accuracy	0.78
Node Classification	Cora	DeepWalk	Accuracy	0.81
Node Classification	Cora	Node2Vec	Accuracy	0.82
Node Classification	Cora	GCN	Accuracy	0.84
Node Classification	Cora	GAT	Accuracy	0.86
Node Classification	Cora	**GCN+GAT (Proposed)**	**Accuracy**	**0.88**
Link Prediction	FB15k-237	TransE	MRR	0.294
Link Prediction	FB15k-237	DistMult	MRR	0.241
Link Prediction	FB15k-237	ComplEx	MRR	0.247
Link Prediction	FB15k-237	**R-GCN (Proposed)**	**MRR**	**0.328**

Figure 11: Summary of Experimental Results

3.5 Discussion

The results presented in this section demonstrate the effectiveness of our proposed hybrid GNN framework for both social network analysis and knowledge graph completion. The GCN+GAT model achieves state-of-the-art performance on the Cora dataset for node classification, outperforming several baseline models. The R-GCN model also achieves state-of-the-art performance on the FB15k-237 dataset for link prediction, demonstrating the power of GNN-based models for knowledge graph completion. Several key insights can be drawn from our experiments. First, the attention mechanism in the GAT layer is crucial for the model's performance, as it allows the model to focus on the most relevant neighbors when aggregating information. Second, the message-passing mechanism in GNNs is highly effective for capturing the structural information of the graph, leading to improved performance compared to traditional machine learning models. Third, the choice of hyperparameters, such as the learning rate and hidden dimensions, can have a significant impact on the model's performance, and careful tuning is necessary to achieve optimal results. One limitation of our study is that we only evaluated our models on two datasets (Cora and FB15k-237). Future work could explore the performance of

ISBN: 978-81-994969-0-3 (Print); 978-81-994969-5-8 (Online)

our models on other datasets and tasks, such as link prediction in social networks and node classification in knowledge graphs. Another limitation is that we only considered a limited set of GNN architectures. Future work could explore the integration of other GNN architectures, such as GraphSAGE and Graph Isomorphism Networks (GINs), into our framework. Despite these limitations, our results provide strong evidence for the effectiveness of GNNs for social network analysis and knowledge graph completion, and our proposed hybrid framework offers a flexible and powerful approach for tackling these important tasks [6].

4. Conclusion

This chapter has provided a comprehensive exploration of Graph Neural Networks (GNNs) and their applications in social network analysis and knowledge graph completion. We began by introducing the foundational concepts of GNNs, including the message-passing mechanism and popular architectures such as Graph Convolutional Networks (GCNs), Graph Attention Networks (GATs), and GraphSAGE. We then presented a detailed review of the relevant literature, covering the key developments in GNN architectures and their applications in social network analysis and knowledge graph completion. The core contribution of this chapter is the proposed hybrid GNN framework, which integrates multiple GNN architectures to address the distinct challenges of social network analysis and knowledge graph completion. For social network analysis, we proposed a GCN+GAT model that combines the power of graph convolution with the attention mechanism to achieve state-of-the-art performance on the Cora citation network dataset for node classification. For knowledge graph completion, we proposed an R-GCN model that uses relation-specific transformations to capture the complex relational information in the FB15k-237 knowledge graph for link prediction. Our experimental results demonstrate the effectiveness of the proposed framework. The GCN+GAT model achieves an accuracy of 88% on the Cora dataset, outperforming several baseline models including MLP, DeepWalk, Node2Vec, GCN, and GAT. The RGCN model achieves an MRR of 0.328 on the FB15k-237 dataset, outperforming traditional embedding models such as TransE, DistMult, and ComplEx. Our ablation study confirms the importance of the attention mechanism in the GCN+GAT model, and our hyperparameter tuning results provide insights into the optimal configuration of the models.

The key findings of this chapter can be summarized as follows:

- **GNNs are highly effective for graph-structured data:** The message-passing mechanism in GNNs allows them to capture the structural information of the graph, leading to improved performance compared to traditional machine learning models.
- **The attention mechanism is crucial for social network analysis:** The GAT

layer with the attention mechanism allows the model to focus on the most relevant neighbors when aggregating information, leading to improved performance on node classification tasks.

- **Relation-specific transformations are crucial for knowledge graph completion:** The R-GCN model with relation-specific transformations can capture the complex relational information in knowledge graphs more effectively than traditional embedding models.
- **Hyperparameter tuning is important:** The choice of hyperparameters, such as the learning rate and hidden dimensions, can have a significant impact on the model's performance, and careful tuning is necessary to achieve optimal results.

Looking forward, there are several promising directions for future research in the field of GNNs. First, the development of more efficient and scalable GNN architectures is crucial for handling large-scale graphs with millions or billions of nodes. Second, the integration of GNNs with other deep learning techniques, such as reinforcement learning and generative models, could lead to new applications and improved performance. Third, the development of interpretable GNN models is important for understanding the decision-making process of the models and building trust in their predictions. Finally, the application of GNNs to new domains, such as drug discovery, protein structure prediction, and financial network analysis, holds great promise for solving real-world problems. In conclusion, this chapter has demonstrated the power and versatility of Graph Neural Networks for social network analysis and knowledge graph completion. The proposed hybrid framework provides a flexible and effective approach for tackling these important tasks, and the experimental results provide strong evidence for the effectiveness of GNNs in capturing the structural information of graph-structured data. As the field of GNNs continues to evolve, we can expect to see even more exciting developments and applications in the years to come.

References

[1] Zonghan Wu et al. "A comprehensive survey on graph neural networks". In: *IEEE transactions on neural networks and learning systems* 32.1 (2020), pp. 4–24.

[2] TN Kipf. "Semi-supervised classification with graph convolutional networks". In: *arXiv preprint arXiv:1609.02907* (2016).

[3] Petar Velickovic et al. "Graph attention networks". In: *stat* 1050.20 (2017), pp. 10–48550.

ISBN: 978-81-994969-0-3 (Print); 978-81-994969-5-8 (Online)

[4] Will Hamilton, Zhitao Ying, and Jure Leskovec. "Inductive representation learning on large graphs". In: *Advances in neural information processing systems* 30 (2017).

[5] Sadamori Kojaku et al. "Network community detection via neural embeddings". In: *Nature Communications* 15.1 (2024), p. 9446.

[6] Antoine Bordes et al. "Translating embeddings for modeling multi-relational data". In: *Advances in neural information processing systems* 26 (2013).

[7] Bishan Yang et al. "Embedding entities and relations for learning and inference in knowledge bases". In: *arXiv preprint arXiv:1412.6575* (2014).

[8] Michael Schlichtkrull et al. "Modeling relational data with graph convolutional networks". In: *European semantic web conference*. Springer. 2018, pp. 593–607.

[9] Allam Balaram et al. "Managing 5G IOT Network Operations and Safety Using Deep Learning and Attention Methods". In: *Wireless Personal Communications* (2024), pp. 1–16.

[10] Mohamad Zamini, Hassan Reza, and Minou Rabiei. "A review of knowledge graph completion". In: *Information* 13.8 (2022), p. 396.

ISBN: 978-81-994969-0-3 (Print); 978-81-994969-5-8 (Online)