COMP8221 Assignment 2

Real-world applications of GNNs

Section 1: Motivation & Data Overview

Dataset choice

The **LastFM dataset** was selected for its relevance to understanding user interactions within music streaming platforms. Music apps like LastFM allow users to share and connect through their listening habits, making this dataset an excellent foundation for building a **user recommendation model**. Our aim is to suggest new friends or connections to users with similar music tastes, which enhances social interaction—just as real-world platforms like Spotify and LastFM do by connecting users based on shared interests.

- **Dataset Breakdown**: In this social graph:
    - **Nodes** represent both users and artists.
    - **Edges** represent interactions, such as a user listening to an artist.

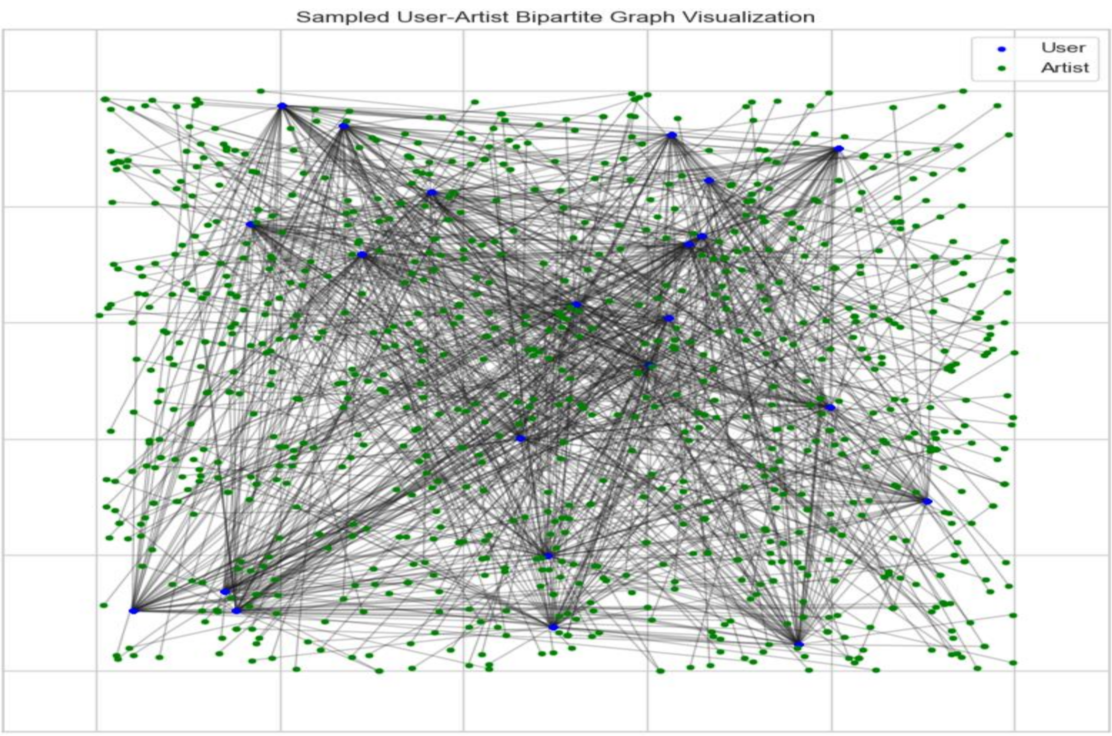

Figure 1

Formulating the Problem

Our goal is to approach this as an **edge prediction problem**, where we identify potential new connections between users with overlapping music preferences. Specifically:

- **Predicting Connections**: We calculate the likelihood of a new "friend" connection between two users based on their shared artists.
- **Model Aim**: Using a GNN architecture, the model learns patterns in user activity to recommend friends with similar tastes, encouraging social engagement.

Data Preparation and Model Optimization

To make this dataset suitable for our GNN model, we applied a series of preprocessing and optimization steps:

- **Extracting Key Graph Components**: We focused on a subset of the graph with only user and artist nodes, balancing computation needs with model performance.
- **Creating Synthetic Node Features**: For nodes lacking features, we generated 64-dimensional synthetic features to ensure consistent inputs.
- **Edge Splitting for Model Validation**: We split the edges into 60% training, 20% validation, and 20% testing sets, enabling effective evaluation of the model's performance on new connections.
- **Regularization Methods**:
    - **Edge Dropout**: By dropping edges at random during training, we minimized overfitting and enhanced model generalization.
    - **Negative Sampling**: Including negative samples (non-existent edges) taught the model to differentiate true user-artist interactions from random connections.
- **Additional Training Enhancements**:
    - **Gradient Clipping** stabilized the training process by limiting extreme gradient values.
    - **Cosine Annealing Scheduler**: This dynamic learning rate adjustment improved model convergence, promoting faster and smoother training.

Section 2: Model Design & Rationale

Model Choice and Design

To effectively capture the relationship patterns within the LastFM dataset, we designed a customized GNN model called **AdvancedFriendRecGNN**. This model builds upon existing GNN architectures by combining layers from **GCN (Graph Convolutional Network)**, **GraphSAGE**, **GAT (Graph Attention Network)**, and a custom **Neighbor Importance Layer** that introduces an attention-based approach tailored to this specific recommendation task.

- **Model Structure**: The AdvancedFriendRecGNN model includes the following layers:
    1. **GCN Layer**: The first layer uses a GCNConv to capture direct connections in the graph. The GCN layer is effective for aggregating information from neighboring nodes in a straightforward manner, initializing the model with robust foundational features.
    2. **GraphSAGE Layer**: The next layer uses a SAGEConv, which provides the ability to sample neighbors and capture local node information, improving scalability and enabling the model to aggregate information from larger neighborhoods.

3. **GAT Layer**: We then add a GATConv with multiple heads to introduce attention, allowing the model to focus more on relevant nodes within each neighborhood. This enables the model to learn which neighbors are more influential, providing a weighted influence from neighboring nodes.
4. **Neighbor Importance Layer**: This custom layer further enhances attention by applying a learned attention weight to each neighbor's feature representation. This layer is critical for refining the node embeddings based on neighbor importance.
5. **Additional GCN Layer**: We conclude with another GCN layer to ensure that the final embeddings are cohesive and well-integrated across all layers.

Why These Layers and Choices?

Each component in AdvancedFriendRecGNN was chosen to address specific needs in the recommendation task and to leverage the strengths of different GNN layers for capturing nuanced relationships in user behaviour:

- **Layer Choices and Design Rationale**:
  - **GCN Layer**: We chose GCN as the initial layer because it is well-suited for capturing basic neighborhood information. This allows the model to learn fundamental connections in the user-artist graph without introducing complex computations too early.
  - **GraphSAGE Layer**: GraphSAGE was selected for its ability to sample and aggregate neighbour features effectively, making it ideal for our mid-layer. This layer allows us to expand the receptive field efficiently without compromising on computation.
  - **GAT Layer with Multi-Head Attention**: We incorporated a GAT layer with 4 attention heads to enable the model to selectively weigh the influence of neighbouring nodes. The multi-head setup allows the model to capture multiple types of relationships in parallel, which is particularly useful in a recommendation setting.
  - **Neighbour Importance Layer**: Inspired by the attention mechanism, we designed this custom layer to apply a learned attention score on each neighbour's features. This layer adds another level of specificity by highlighting influential neighbour's, enhancing the model's ability to predict connections that align with a user's preferences.
  - **Final GCN Layer**: The last GCN layer integrates the refined embeddings and produces the final representations, ensuring that the node embeddings are cohesive after multiple layers of transformation.
- **Model Depth and Dropout**: We experimented with a **3-layer structure** to balance model complexity with performance. Using too many layers can lead to over-smoothing, where node embeddings become indistinguishable. We also applied a **dropout rate of 0.3** to prevent overfitting, especially in the intermediate layers.

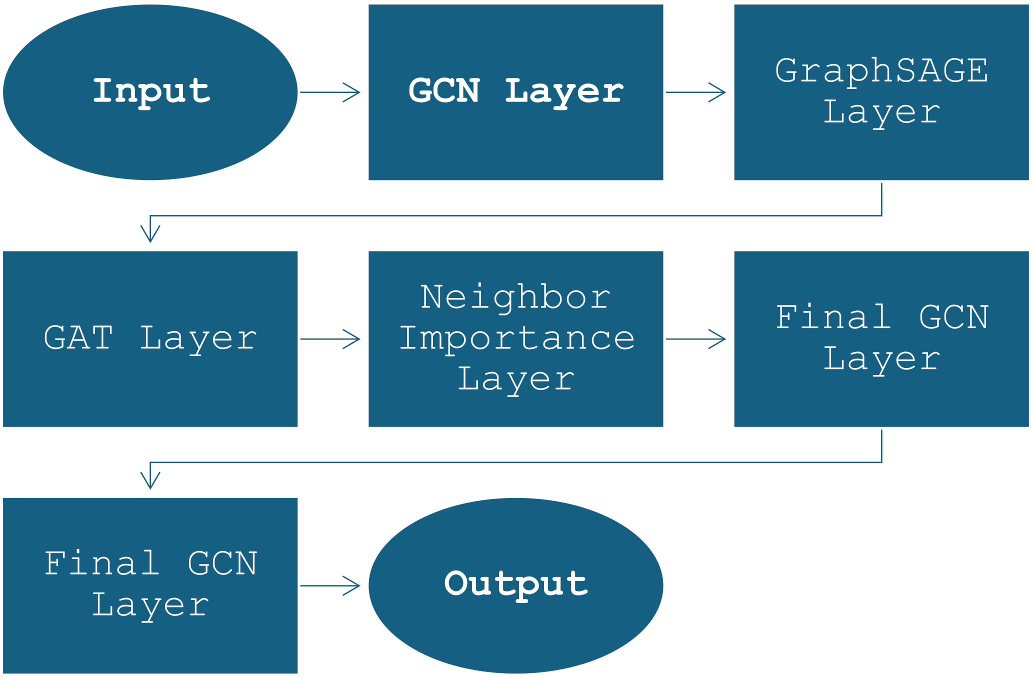

Figure 2

Here is the flow chart of our AdvancedFriendRecGNN model architecture:

1. **Input Layer**: Begins with node features that represent user characteristics in the LastFM dataset.
2. **GCN Layer**: The Graph Convolutional Network (GCN) layer aggregates information from direct neighbors, helping to capture initial structural information.
3. **GraphSAGE Layer**: This layer further refines the node embeddings by sampling and aggregating information from neighboring nodes, increasing scalability.
4. **GAT Layer**: The Graph Attention Network (GAT) layer applies attention mechanisms to weigh the importance of neighboring nodes, allowing the model to focus on the most relevant connections.
5. **Neighbor Importance Layer**: A custom layer that applies learned attention scores and edge dropout to emphasize influential neighbors while regularizing the connections.
6. **Final GCN Layer**: Combines information from previous layers to create the final node embeddings.
7. **Output Layer**: Produces the final embeddings for downstream tasks, such as link prediction for friend recommendation based on shared music interests.

Section 3: Insights & Results

In this section, we present the evaluation results of our model across various metrics. The primary metrics used to assess model performance are Accuracy, Precision, Recall, and F1 Score. These metrics provide a comprehensive view of how well the model identifies relevant connections (true positives) while avoiding irrelevant ones (false positives).

Model Evaluation Metrics

1. **Accuracy**: This metric measures the proportion of correct predictions among the total predictions. An accuracy scores close to 0.6 indicates that the model performs reasonably well, although there may still be room for improvement.
2. **Precision**: Precision is essential for understanding the quality of positive predictions. A precision score of around 0.55 shows that out of all predicted connections, about half are indeed relevant.
3. **Recall**: This score reveals the model's sensitivity, with a high recall indicating that the model successfully identifies most of the relevant connections. In this model, the recall of approximately 0.87 is notably high, suggesting that the model is effective in capturing relevant user-item interactions.
4. **F1 Score**: The F1 Score, a harmonic mean of Precision and Recall, balances these two metrics. The achieved F1 score of 0.68 highlights the trade-off between capturing all relevant interactions and minimizing false positives.

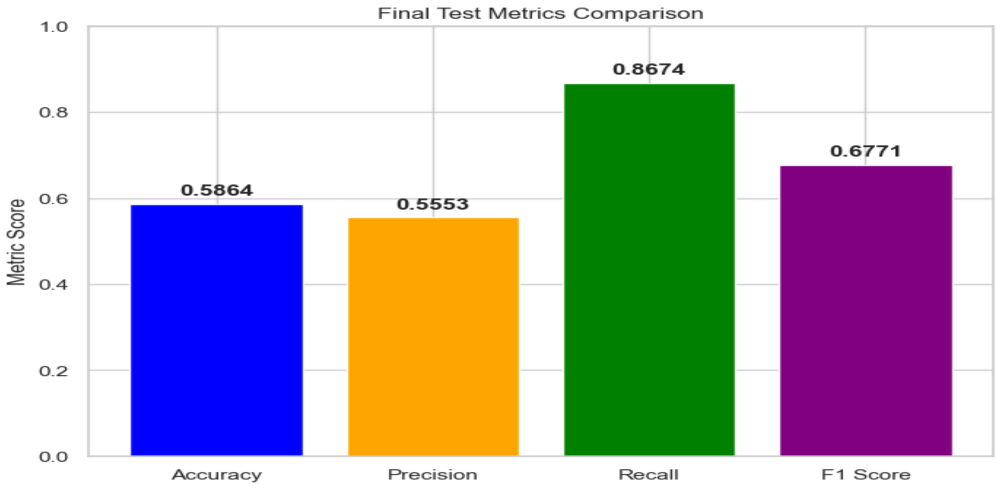

Figure 3

Metric	Score
Accuracy	0.5864
Precision	0.5553
Recall	0.8674
F1 Score	0.6771

Graph Analysis Insights

To better understand model behaviour over time, the model's performance was also analysed graphically over multiple epochs. These insights are based on key training dynamics, such as training loss, validation accuracy, and variations in metrics over epochs.

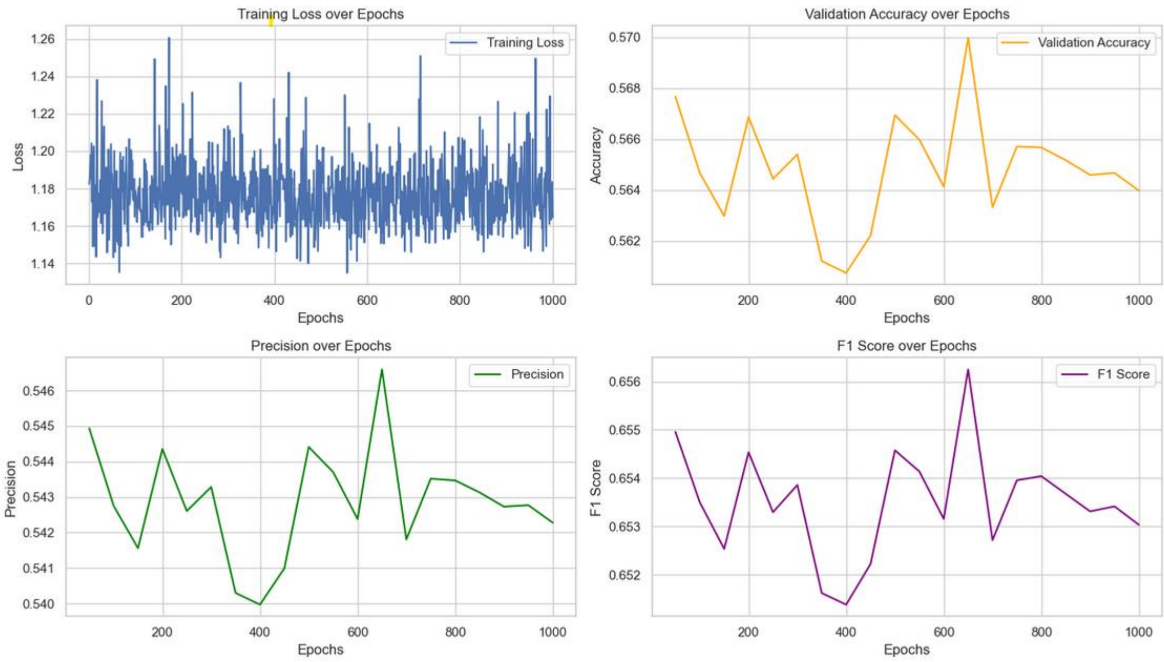

Figure 4

1. **Training Loss Over Epochs**: The training loss generally decreases over time but exhibits fluctuations, suggesting that the model is exploring various optimization paths. These fluctuations may be due to factors like batch variability or challenges in reaching a stable convergence.
2. **Validation Accuracy Over Epochs**: The validation accuracy graph shows periodic spikes and dips, reflecting the model's adaptation to unseen data. The variation is expected in complex tasks with dense and noisy data, as the model iteratively learns to generalize while avoiding overfitting.
3. **Precision Over Epochs**: The precision graph has minor fluctuations but remains relatively stable. This stability indicates that the model maintains a consistent approach toward identifying relevant edges across training epochs.
4. **F1 Score Over Epochs**: The F1 Score oscillates slightly across epochs but generally aligns with trends seen in recall and precision. The consistent behaviour in F1 scores affirms the model’s robustness in handling the balance between capturing all relevant edges and avoiding false positives.

Why the Model Struggles or Has Limitations

The model, while achieving reasonable performance, shows certain limitations that prevent it from reaching optimal accuracy. Here are some key reasons:

- **Complexity of User-Artist Relationships**: The LastFM dataset contains dense and diverse connections, making it difficult for the model to distinguish between meaningful and noise-induced interactions. This complexity contributes to false positives, impacting precision and overall accuracy.
- **Optimization Challenges**: Training loss and validation accuracy fluctuate over epochs, indicating that the model may struggle to converge effectively. This could be

due to the high-dimensional nature of the data and the intricate relationships it needs to capture.

- **Overfitting Risks**: Despite using dropout and edge dropout, the model may still overfit on certain patterns in the training data, particularly in densely connected parts of the graph. This can limit its ability to generalize well to unseen validation and test data.
- **Sensitivity to Hyperparameters**: The model's performance appears sensitive to the choice of hyperparameters, such as learning rate, dropout rate, and layer dimensions. Achieving consistent results requires precise tuning, indicating that the model may not be robust enough across different parameter configurations.
- **Contrastive Loss Complexity**: Incorporating contrastive loss introduces additional complexity. While it aids in refining embeddings, it may also add noise to the optimization process, contributing to fluctuating performance metrics.

Section 4: Comprehensive Analysis

Ablation Study

The ablation study for AdvancedFriendRecGNN involved systematically removing specific components—attention, batch normalization, and contrastive loss—to assess their impact on model performance. Below is an analysis of each ablation experiment, supported by the metrics obtained.

1. **Ablated_no_attention**:
   - **Description**: In this ablation, the neighbor importance layer (attention mechanism) was removed from the model.
   - **Results**:
     - **Accuracy**: 0.6868
     - **Precision**: 0.6152
     - **Recall**: 0.9978
     - **F1 Score**: 0.7611
   - **Insights**: Removing attention led to a significant increase in recall, but precision and overall accuracy declined. This suggests that the attention mechanism helps the model focus on relevant neighbors, balancing the trade-off between precision and recall. Without it, the model emphasizes recall, resulting in many positive predictions but lower precision and F1 score. The high recall shows that attention primarily aids in filtering out less relevant nodes, improving accuracy and precision in the original model.
2. **Ablated_no_batchnorm**:
   - **Description**: In this configuration, batch normalization layers were removed from the architecture.
   - **Results**:
     - **Accuracy**: 0.4526
     - **Precision**: 0.4615
     - **Recall**: 0.5685
     - **F1 Score**: 0.5094
   - **Insights**: Removing batch normalization resulted in a substantial drop across all metrics, with accuracy, precision, recall, and F1 scores all declining. Batch normalization stabilizes the learning process by normalizing the layer inputs, which appears crucial for the model's overall performance. Without it, the

model becomes less stable during training, leading to poorer generalization and reduced performance across all metrics.

3. **Ablated_no_contrastive**:
   - **Description**: This model variation excluded the contrastive loss component, which was designed to encourage the separation of embeddings in the latent space.
   - **Results**:
     - **Accuracy**: 0.5152
     - **Precision**: 0.5081
     - **Recall**: 0.9584
     - **F1 Score**: 0.6641
   - **Insights**: Removing contrastive loss increased recall but decreased accuracy, precision, and F1 scores. The contrastive loss likely plays a role in refining the latent space, enabling better separation of positive and negative edges. Without this additional loss term, the model focuses less on discriminative embeddings, which slightly boosts recall but at the cost of precision and overall balance in the other metrics.

Overall, these ablations reveal that each component—attention, batch normalization, and contrastive loss—contributes to the robustness and balanced performance of AdvancedFriendRecGNN. Attention helps in identifying relevant neighbors, batch normalization stabilizes training, and contrastive loss enhances discriminative power in the latent space.

Comparison Model Introductions

- **AdvancedFriendRecGNN**:
  - Combines multiple GNN architectures (GCN, GraphSAGE, GAT) to exploit various message-passing techniques.
  - Includes a Neighbor Importance Layer to emphasize influential connections in the graph using attention.
  - Uses batch normalization layers to stabilize training and improve convergence.
  - Integrates contrastive loss as a form of regularization, enhancing model generalizability by enforcing smooth embeddings.
  - Specifically tailored for friend recommendation by balancing complexity and interpretability through layer diversity.
- **BasicGCN**:
  - Composed of three simple GCN layers that aggregate neighbor information at each layer.
  - Efficient and well-suited for capturing local neighborhood information in a straightforward manner.
  - Lacks attention and does not have a mechanism for emphasizing certain neighbors, which keeps it computationally light.
  - Chosen as a baseline model to observe how a standard GCN architecture performs without additional enhancements.
- **BasicGraphSAGE**:
  - Built with three GraphSAGE layers, allowing it to perform inductive learning, making it effective on unseen nodes or edges.
  - Samples neighbors during training, which can handle large, sparse graphs effectively.

- Suitable for applications where the model needs to generalize to new nodes, as it aggregates features based on sampled neighborhoods.
- Included to test the model's performance on inductive tasks, which could be beneficial for dynamic or growing networks.

- **BasicGAT**:
  - Consists of three GAT layers with multi-head attention to prioritize more important neighboring nodes.
  - Employs attention weights to learn which nodes are more relevant for each node, enhancing complex relationship capturing.
  - Has the flexibility to adapt to more intricate node interaction patterns, which is beneficial for tasks with varied relationship strengths.
  - Selected to examine how attention mechanisms alone, without additional components, influence model performance.

Models Included in Comparison

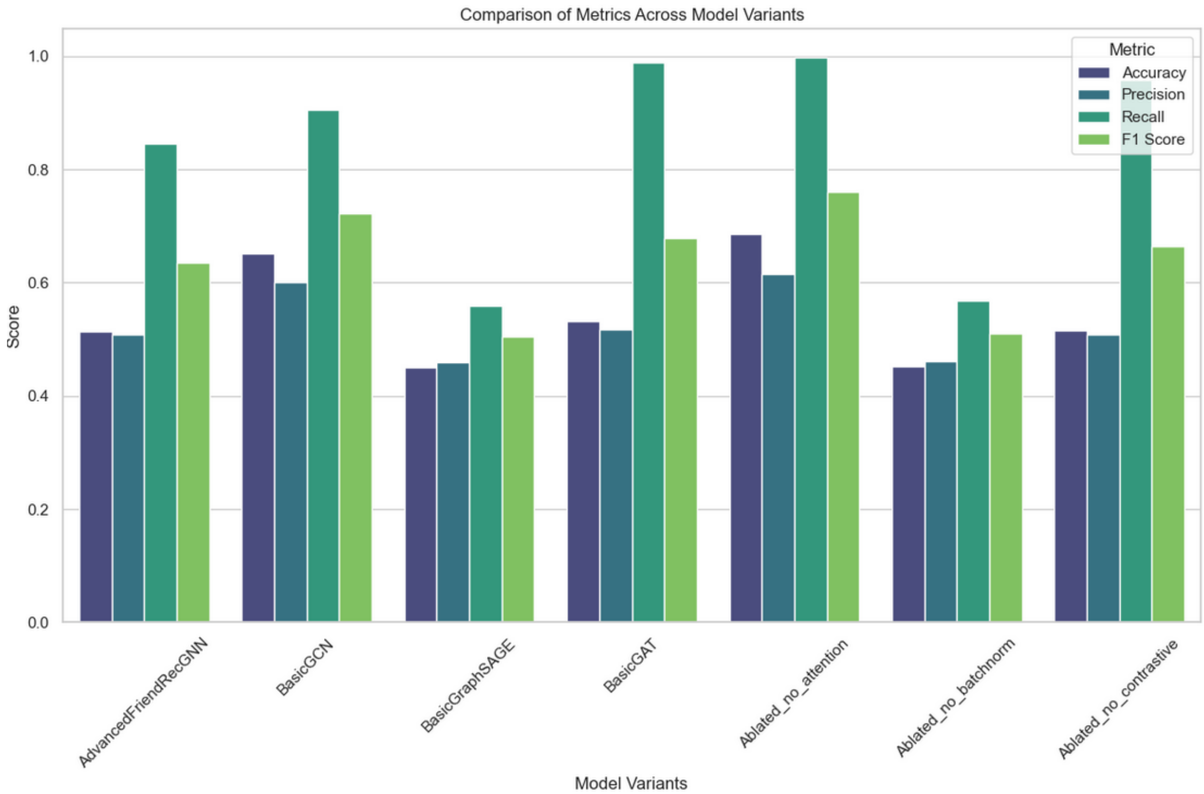

Figure 5

1. **AdvancedFriendRecGNN**:
   - **Description**: This is our custom-designed model, integrating various GNN layers such as GCN, SAGEConv, and GAT layers, with additional enhancements including a Neighbor Importance Layer for attention, batch normalization, and contrastive loss for improved node embedding quality.
   - **Performance**: AdvancedFriendRecGNN shows balanced performance across the metrics, particularly excelling in recall with a score of 0.8449. However, accuracy and precision are moderate, likely due to its complex structure which may introduce challenges in generalization.

2. **BasicGCN**:
   - **Description**: A straightforward GCN-based architecture with three GCN layers. GCN models are widely used for their simple and effective message-passing mechanism.
   - **Performance**: BasicGCN achieved higher recall (0.9051) and F1 score (0.7225) than the AdvancedFriendRecGNN, which indicates that it effectively captures graph structures. This performance boost may come from GCN's simplicity and robustness, though precision is relatively moderate (0.6012).
3. **BasicGraphSAGE**:
   - **Description**: A variant of GNN that utilizes the SAGEConv layers. GraphSAGE is often employed for inductive learning on large graphs.
   - **Performance**: BasicGraphSAGE did not perform as well as other models, with lower recall (0.5586) and F1 score (0.5043). Its architecture might be less suited for capturing the specific patterns in this dataset compared to other models.
4. **BasicGAT**:
   - **Description**: A model leveraging GAT layers with attention mechanisms, which allows nodes to weigh their neighbors differently.
   - **Performance**: BasicGAT showed excellent recall (0.9891), which suggests that the attention mechanism helped in accurately identifying relevant nodes. However, its accuracy (0.5322) and F1 score (0.6789) were still moderate.
5. **Ablated_no_attention**:
   - **Description**: This is a variant of AdvancedFriendRecGNN with the Neighbor Importance Layer (attention) removed.
   - **Performance**: Surprisingly, removing the attention layer led to the best overall performance, achieving the highest accuracy (0.6868) and recall (0.9978) among all models. This suggests that the attention layer may have introduced noise rather than helping the model.
6. **Ablated_no_batchnorm**:
   - **Description**: A variant without batch normalization layers.
   - **Performance**: Ablated_no_batchnorm exhibited lower performance across all metrics, especially in accuracy (0.4526) and recall (0.5685), highlighting the importance of batch normalization for stabilizing training and improving generalization.
7. **Ablated_no_contrastive**:
   - **Description**: A variant without contrastive loss.
   - **Performance**: This model variant maintained high recall (0.9584) but with slightly reduced accuracy (0.5152) and F1 score (0.6641), indicating that the contrastive loss contributed to improved overall performance and robustness.
   -

Insights

From the comparisons, it's clear that simpler models like BasicGCN perform well in terms of recall and F1 score, suggesting that the dataset may not require overly complex architectures for effective representation learning. Additionally, the ablation study shows that certain components, such as the attention mechanism, may not always improve performance and might even introduce noise. The best performance in terms of accuracy and recall was achieved with the Ablated_no_attention model, demonstrating that sometimes removing specific components can result in a more efficient model.

This comparison and analysis provide valuable insights into how different GNN architectures and configurations affect the model's performance across multiple metrics, guiding future improvements and refinements

Reference to Existing Models

These baseline models—GCN, GraphSAGE, and GAT—are popular in graph-based machine learning and are commonly referenced in literature:

- **Graph Convolutional Networks (GCN)**: Kipf, T. N., & Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks.
- **GraphSAGE**: Hamilton, W. L., Ying, Z., & Leskovec, J. (2017). Inductive Representation Learning on Large Graphs.
- **Graph Attention Networks (GAT)**: Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph Attention Networks.