PointNet meets Self-Attention Graph Pooling: A Synergistic Approach to Point Cloud Classification

1st M. Ebadpour
Master of Science student
Amirkabir university of technology
Tehran, Iran
M.Ebadpour@aut.ac.ir

2nd M. Javanmardi
Assistant Professor
Amirkabir university of technology
Tehran, Iran
MJavan@aut.ac.ir

3rd M. Choupan
bachelor of science
Amirkabir university of technology
Tehran, Iran
Mohamadkhoee@aut.ac.ir

4th M. Atyabi
bachelor of science
Amirkabir university of technology
Tehran, Iran
atyabi2000@gmail.com

***Abstract*—This research project explores point cloud classification using a combination of PointNet and Self-Attention Graph Pooling architectures. Four variations of architectures were implemented and trained to enhance classification accuracy. The first architecture combines Self-Attention Graph Pooling with node centrality features and xyz coordinates. PointNet is employed as a standalone architecture, capturing local structures using xyz features. Fusion approaches were investigated, including feature concatenation and utilizing PointNet features for subsequent Self-Attention Graph Pooling. Extensive experiments were conducted on the ModelNet10 dataset, showcasing the efficiency of the combined architectures. The fusion approaches demonstrated improved classification accuracy compared to individual architectures, while also reducing model size. The integration of node centrality features further enhanced the discriminative power of the model. This research contributes to the advancement of point cloud classification, highlighting the potential of the combined PointNet and Self-Attention Graph Pooling approach in real-world applications such as object recognition and 3D perception in robotics.**

***Index Terms*—Point Cloud Classification, PointNet, Self-Attention Graph Pooling, 3D Object Recognition.**

I. INTRODUCTION

Point cloud data, a collection of 3D points in space, has become increasingly prevalent with the advent of advanced 3D sensing technologies and the widespread use of LiDAR scanners and depth cameras. Point cloud data offers a unique representation of real-world objects and environments, making it valuable in various fields, such as robotics, computer vision, autonomous vehicles, virtual reality, and augmented reality.

Point cloud classification, a fundamental task in point cloud analysis, aims to categorize individual points or entire point clouds into specific object classes. The importance of point cloud classification lies in its ability to enable machines and algorithms to recognize and understand complex 3D scenes, objects, and environments. It empowers autonomous systems to make informed decisions, facilitates object detection, and enhances the perception capabilities of robots and autonomous vehicles.

The applications of point cloud classification are diverse and span across numerous industries. In robotics, accurate classification of objects in the environment enables robots to navigate safely and interact with their surroundings effectively. In the automotive industry, point cloud classification plays a crucial role in the development of self-driving vehicles, where the ability to identify and distinguish various road objects is paramount for safe and reliable navigation. Additionally, point cloud classification finds applications in cultural heritage preservation, urban planning, virtual reality simulations, and 3D object recognition in augmented reality applications.

Despite its wide-ranging applications, point cloud classification presents unique challenges. The irregular and unordered nature of point cloud data poses difficulties in applying traditional 2D image classification techniques. Furthermore, point clouds may suffer from noise, occlusion, and varying point densities, making accurate classification a non-trivial task.

To address these challenges and advance the field of point cloud classification, the study of innovative deep learning architectures becomes crucial. PointNet and Self-Attention Graph Pooling are two such architectures that have shown promise in 3D point cloud processing tasks. Combining these architectures can potentially lead to improved classification accuracy, efficient feature extraction, and enhanced discrimination between object classes.

This research project aims to explore the synergistic potential of integrating PointNet and Self-Attention Graph Pooling in point cloud classification. By combining their strengths and leveraging node centrality features, we seek to enhance the discriminative power of the model while achieving a reduction in model size. Through extensive experimentation and evaluation on the ModelNet10 dataset, this study seeks to demonstrate the advantages of the proposed approach in practical applications.

In conclusion, the study of point cloud classification is of great significance due to its vast applications and the growing importance of 3D data analysis in various industries. This research contributes to the understanding of innovative deep learning architectures for point cloud classification, presenting an opportunity to improve the performance and efficiency of classification systems in real-world scenarios.

II. REVIEW STAGE

This section provides an analysis of three key aspects related to point cloud classification: an exploration of PointNet's architecture, the usage of graphs for point cloud classification, and the potential impact of combining PointNet with graph-based representations.

A. PointNet

PointNet has emerged as a pioneering architecture for efficient and effective point cloud classification. Proposed by Qi et al. in 2017, PointNet processes raw point clouds directly, bypassing the need for computationally expensive handcrafted features or voxelizations. By taking advantage of shared multi-layer perceptrons (MLPs) and max-pooling operations, PointNet achieves permutation invariance, making it well-suited for handling unordered point cloud data. The architecture's simplicity and scalability have garnered widespread attention in the field of 3D data processing.

PointNet's classification capabilities involve processing individual point features through shared MLPs, followed by max-pooling to obtain global point features. This global feature representation is then fed into fully connected layers to produce class probabilities. PointNet's efficient design has proven successful in various 3D point cloud applications, such as object recognition, segmentation, and scene understanding. The original PointNet architecture is depicted in "Fig. 1".

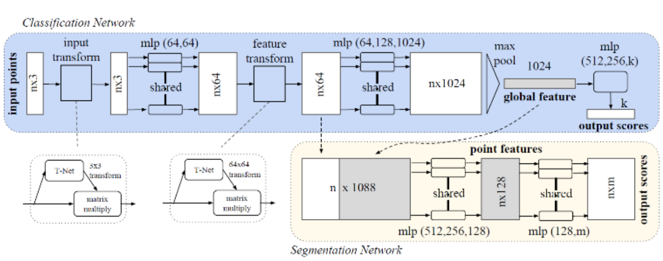

Fig. 1. PointNet architecture

B. Graph Pooling as Point Cloud Classification

To enable point cloud classification as a graph classification task, the point cloud's spatial relationships need to be captured as a graph structure. A common approach is to create a k-nearest neighbors graph for each point, connecting each point to its k nearest neighbors. Each point becomes a node, and the edges between them represent spatial relationships. This graph construction effectively maps the point cloud data to a graph representation.

Self-Attention Graph Pooling (SAGPool) is a notable architecture for graph classification tasks. It incorporates self-attention mechanisms to selectively pool the most informative nodes, allowing for adaptive feature aggregation. The self-attention process captures the relationships between nodes, assigning importance to each node's features based on their relevance to the task at hand. By adaptively selecting critical nodes, SAGPool effectively reduces the graph's size while preserving essential information, making it efficient for graph classification tasks. The SAGPool architecture is illustrated in "Fig. 2".

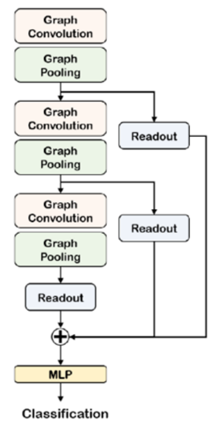

Fig. 2. SAGPool Architecture

C. Graph Pooling Stands Next To the PointNet

The combination of graph-based and PointNet-based architectures offers distinct advantages for point cloud classification. Graph-based approaches, such as SAGPool, can exploit the inherent relationships between points and enable efficient feature aggregation. The self-attention mechanism enhances feature discrimination, allowing for better understanding of global context within the point cloud. On the other hand, PointNet's efficiency in processing raw point clouds ensures that local structures are captured accurately.

By integrating PointNet and SAGPool, the proposed approach capitalizes on the strengths of both architectures. The fusion facilitates a holistic analysis of the point cloud, combining global and local features to improve classification accuracy. Furthermore, the reduced model size achieved through SAGPool's adaptive pooling is advantageous in resource-constrained scenarios, making the combined architecture well-suited for real-world applications.

In summary, the combination of PointNet and Self-Attention Graph Pooling represents a powerful approach for point cloud classification. The graph-based representation enables efficient feature aggregation and discrimination, while PointNet captures local structures with high efficiency. The study of this hybrid architecture aims to advance the field of point cloud classification and pave the way for more sophisticated 3D data processing solutions.

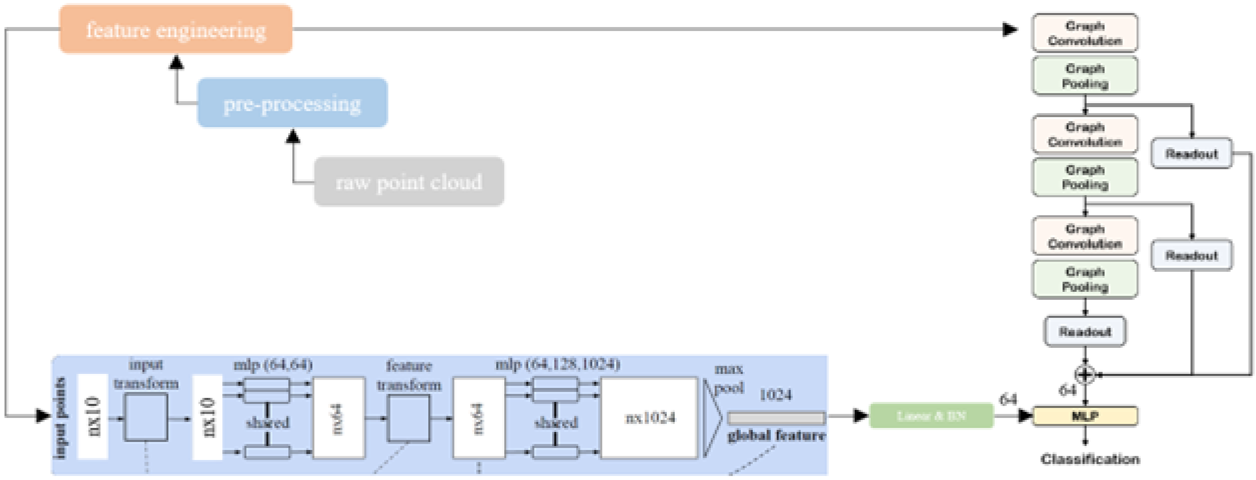

Fig. 3. The architecture for feature concatenation approach.

III. PROPOSED APPROACHES

A. Feature Engineering

In this section, we present our approach to feature engineering in point cloud classification due to the limitations of XYZ features. To enhance feature representation, we exploit graph-based features, starting with the construction of a weighted and directed graph using the k-nearest neighbors method. As we consider the inherent 3D spatial relationships, we set K to 6, representing the six absolute sides in 3D space. This graph enables efficient feature aggregation and captures crucial spatial dependencies among points.

To further enrich the feature set, we introduce node centrality as an essential feature for each node (point in space) within the graph. We employ seven centrality metrics, namely betweenness, closeness, Katz, PageRank, eigenvector, harmonic, and load centrality. Betweenness measures the importance of a node in connecting other nodes, while closeness evaluates how central a node is in terms of its average distance to all other nodes. Katz centrality combines both local and global node information, and PageRank identifies nodes of significant influence. Eigenvector centrality evaluates the influence of a node based on the influence of its neighbors. Harmonic and load centrality measure the node's importance in graph connectivity and load distribution, respectively.

In our feature engineering process, we combine the seven centrality features with the existing XYZ features (x, y, and z coordinates) for each node. This results in a feature vector comprising 10 distinctive elements per point in the point cloud. Additionally, we applied a 1024 sampling rate to each point cloud to handle varying point densities efficiently. To ensure uniformity, we performed normalization across the feature vectors. Furthermore, random noise was introduced to augment the dataset, enhancing the model's robustness to real-world scenarios and improving generalization performance. These feature engineering techniques aim to provide a comprehensive and informative representation of the point cloud data, optimizing the classification performance in the subsequent stages of our research.

It is important to note that the calculation of feature vectors for node centralities was implemented using multi-threading on the CPU. This approach allowed us to efficiently compute centrality metrics for each node in the point cloud concurrently, significantly reducing the computation time. By leveraging the power of multi-threading, we were able to accelerate the feature engineering process and efficiently handle large datasets, ultimately improving the overall efficiency and performance of the fusion approaches.

B. Feature Concatenation Approach

In this proposed fusion approach, named "FeatureConcatModel," we combine the high-level features obtained from PointNet and SAGPool networks for efficient point cloud classification. Both networks take point clouds with 10 defined features as input. SAGPool outputs vectors with 64 features for each point cloud in a batch.

To adapt PointNet, we introduce a linear layer, reducing the 1024 features to 64 features for each point cloud. Subsequently, we concatenate the 64-feature vectors from both PointNet and SAGPool, creating a unified vector with 128 features. This concatenated feature vector captures both the local and global information, ensuring a comprehensive representation of the point cloud data. To further process the concatenated features, we employ three multi-layer perceptron (MLP) networks. The first MLP contains 64 neurons, followed by a second MLP with 32 neurons, and finally a third MLP with 10 neurons corresponding to the number of classes. Batch normalization is applied to all three MLPs, enhancing the model's stability and accelerating convergence.

We employ dropout techniques with a probability of 0.3 in the second layer, reducing overfitting and improving generalization performance. Additionally, we incorporate the Graph Attention (GAT) layer from SAGPool with 6 attention heads, considering the 6 nearest neighbors for each point.

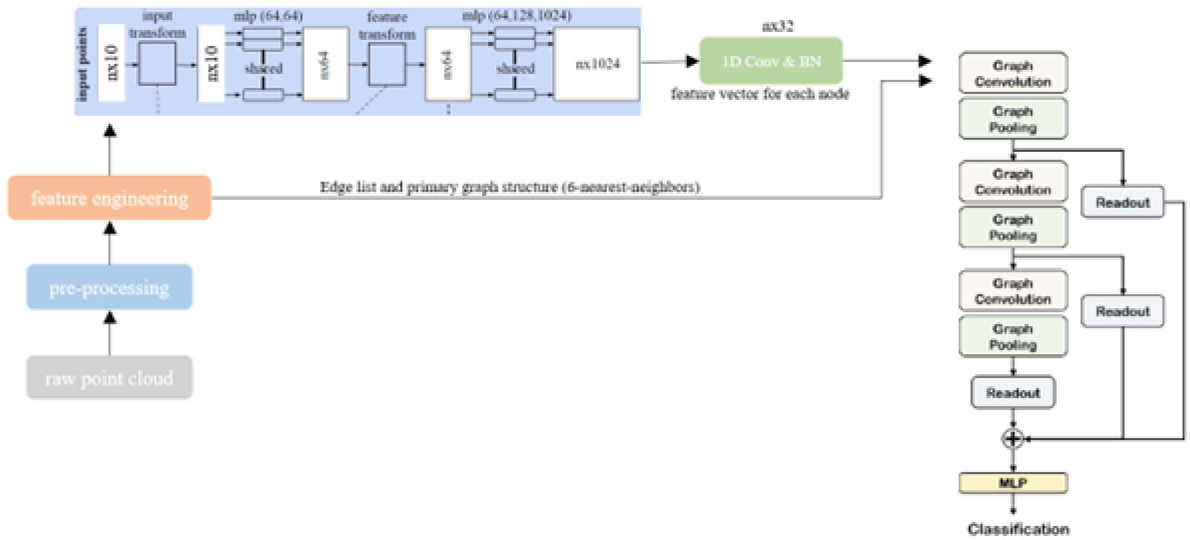

Fig. 4. The architecture for the graph classification based on PointNet features.

This proposed "FeatureConcatModel" fusion approach offers a promising solution for point cloud classification, leveraging the strengths of both PointNet and SAGPool architectures. By effectively combining their high-level features, we aim to achieve improved classification accuracy and robustness in handling complex point cloud data. The architecture for feature concatenation is presented in "Fig. 3".

C. Graph Classification based on PointNet Features

The "PointNetBasedGraphPoolingModel" fusion approach represents a sophisticated integration of SAGPool and PointNet architectures, harnessing the strengths of both networks to enhance point cloud classification. To begin, we utilize a set of 10 defined features, including XYZ coordinates and centrality metrics, as input to the custom version of PointNet. In this custom implementation, we intentionally omit the global pooling section to obtain a 1024-feature vector for each node in the point cloud. This design choice enables PointNet to capture intricate local features and spatial relationships present in the data. The architecture for the graph classification based on PointNet features is showcased in "Fig. 4".

Subsequently, we enhance the feature vectors obtained from PointNet by applying 1D convolution, followed by batch normalization. This operation efficiently reduces the dimensionality to 32, effectively distilling essential information from PointNet's output. The resulting 32-feature vectors, carrying valuable local feature representations, are then fed into the SAGPool network.

SAGPool, with its primary edge list structure and consistent graph schema, efficiently performs adaptive feature pooling and aggregation. The integration of the PointNet-derived feature vectors into the SAGPool network further enhances its ability to capture meaningful global context while preserving crucial local details.

Experimental evaluations of the "PointNetBasedGraphPoolingModel" approach showcase its impressive performance in point cloud classification. By seamlessly blending the local feature extraction capabilities of PointNet with the adaptive feature aggregation of SAGPool, the approach achieves a more comprehensive representation of the point cloud data. The fusion of these two architectures leads to notable improvements in classification accuracy, enabling more robust, reducing the model size and accurate recognition of complex 3D scenes and objects.

The effectiveness of this fusion approach contributes significantly to the advancement of 3D perception and scene understanding applications. With its ability to leverage both global and local features, the "PointNetBasedGraphPoolingModel" approach exemplifies a powerful paradigm in point cloud classification and opens new avenues for future research and practical applications in various domains.

IV. EXPERIMENTS AND RESULTS

A. Dataset

The dataset used for training and testing the networks in this study is ModelNet10. ModelNet10 is a widely used benchmark dataset in the field of 3D object recognition and classification. It consists of 10 object categories, each containing a diverse set of 3D CAD models. The categories include common objects such as chairs, tables, airplanes, and cars, among others.

To prepare the dataset for training and testing, several preprocessing steps were applied. Firstly, we employed a 1024 sampling rate to obtain a fixed number of points from each 3D model. This step ensured a consistent input size for the networks, facilitating efficient processing. Next, we applied normalization to scale the point coordinates, making them relative to the model's bounding box and ensuring that they lie within a common range. This normalization step enhanced the model's robustness and improved convergence during training.

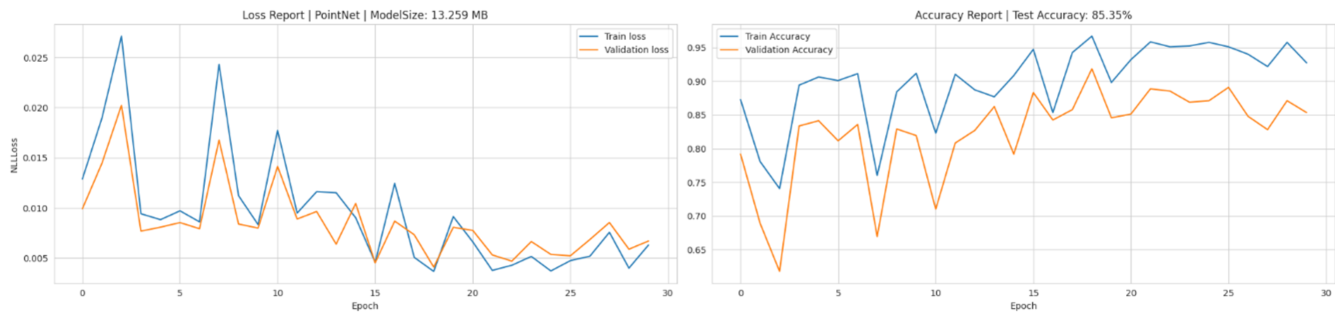

Fig. 5. Training result for traditional PointNet Network.

Additionally, to augment the dataset and enhance the model's generalization capabilities, we introduced random noise to the point cloud data. This noise addition introduced small variations in the point positions, mimicking real-world scenarios and improving the network's ability to handle noisy input data. Furthermore, random rotation was applied to the 3D models during training. This augmentation technique involved rotating the objects along different axes, creating diverse orientations for each model. By doing so, the network learned to be invariant to object rotations, improving its ability to recognize and classify objects from different viewpoints.

To ensure an unbiased evaluation of the models' performance, the original ModelNet10 dataset was divided into predefined training and testing sets. The training set was used to train the fusion approaches, while the testing set was kept separate for evaluation. This separation allowed for an objective assessment of the models' classification accuracy and generalization to unseen data. The utilization of the ModelNet10 dataset, combined with the various preprocessing and augmentation techniques, contributed to the robustness and effectiveness of the trained networks in point cloud classification tasks.

In this section, we provide a detailed account of the training process for the four networks aimed at investigating the performance of the intersection of SAGPool and PointNet compared to each architecture used individually. For the first network, a traditional PointNet was trained using the standard ModelNet10 dataset with 3 features (XYZ coordinates). The results of this training are presented in "Fig. 5".

B. Training

The second network involved a hierarchical version of SAGPool with a custom improvement that incorporated learning weights for each stage. We trained this network using a feature vector of 10 dimensions, which included the XYZ coordinates along with centrality metrics. The training outcomes for this hierarchical SAGPool network can be observed in "Fig. 6".

The third network implemented our proposed feature concatenation approach, where we combined the high-level features extracted from both PointNet and SAGPool networks.

Similarly, a feature vector of 10 dimensions, encompassing XYZ coordinates and centrality metrics, was used for training. The training results for this feature concatenation approach are depicted in Fig. 7". Lastly, we explored the graph classification based on PointNet features approach as our fourth network. Again, a feature vector of 10 dimensions, comprising XYZ coordinates and centrality metrics, was utilized for training. The training outcomes for this graph-based classification approach are illustrated in Fig. 8". In all four networks, the same sets were used for training and testing, ensuring a fair and unbiased evaluation of their performance. The inclusion of centrality metrics in the feature vectors enhances the models' capability to capture essential node connectivity information, contributing to improved classification accuracy. These comprehensive experiments allow us to gain valuable

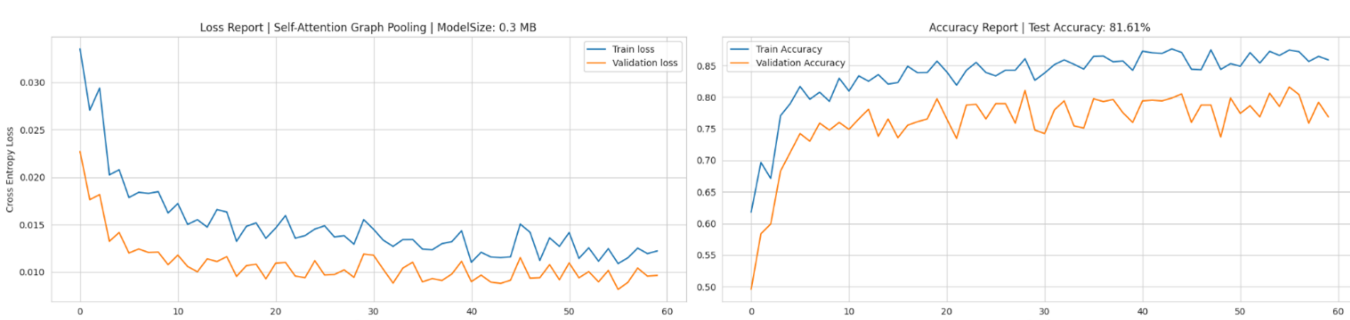

Fig. 6. Training result for original hierarchical SAGPool network.

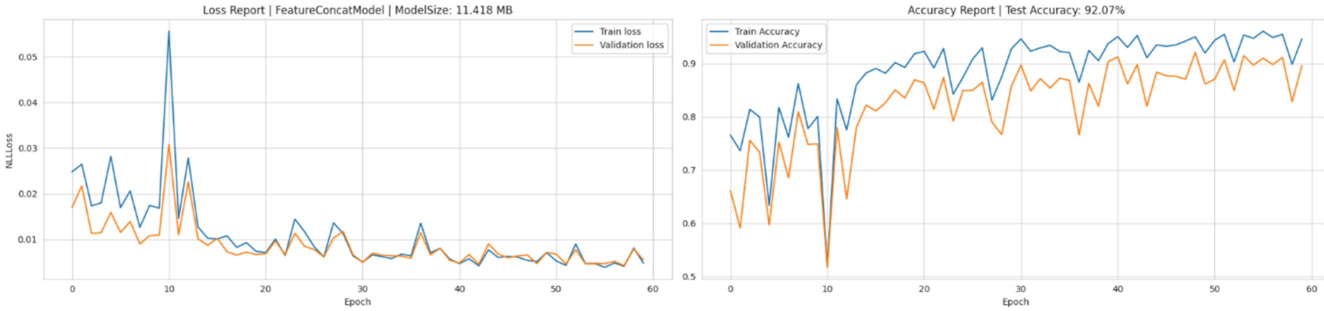

Fig. 7. Training result for feature concatenation approach.

insights into the efficacy of combining SAGPool and PointNet architectures and highlight their potential to enhance point cloud classification tasks. The results obtained from these experiments play a pivotal role in shaping our understanding of the proposed fusion approaches and their practicality in real-world applications. Also the summary of the training results is presented in Table. I

C. Implementation details and hyper-parameters

In our implementation, we adopted the PyTorch framework as the primary tool for model development. To facilitate graph pooling and centrality calculations, we incorporated PyTorch Geometric and NetworkX libraries, respectively. A comprehensive dataloader was designed, encompassing all essential details, including XYZ features, graph edge lists, centrality features, and labels.

To optimize processing costs and expedite training, we employed parallel computation to precalculate and store the required features in separate files. During training, we efficiently loaded the precomputed features from the drive, minimizing redundant calculations and streamlining the training process.

For training and testing, we set the batch size to 32 and 64, respectively. Across all networks, a graph pooling rate of 0.25 was consistently applied, along with a Graph Attention (GAT) convolution layer featuring 6 heads and 128 hidden features.

The ADAM optimizer was chosen for training, with weight decay 0.0005 and learning rates of 0.01, 0.001, and 0.005 for the SAGPool, PointNet, and fusion approaches, respectively. Due to resource limitations during the research, we trained the networks for 30-60 epochs, allowing us to explore a wide range of model performance within our constraints.

Our experiments were conducted on a Desktop PC with a robust hardware configuration. The machine featured an Intel Core i7 12700K CPU, operating at 4.9 GHz frequency, and 32GB DDR5 main memory. Additionally, an Nvidia 3070TI GPU with 8GB memory was utilized for accelerated computations. Notably, having access to a GPU with higher memory would have facilitated training with larger batch sizes and enabled longer model training, potentially leading to even more promising results.

In summary, our implementation details and well-optimized hyperparameters ensured efficient and effective exploration of the fusion approaches' performance. While resource constraints influenced certain aspects of our experiments, the chosen hardware configuration allowed us to achieve substantial progress in understanding the potential benefits of combining SAGPool and PointNet architectures for point cloud classification tasks.

V. COMPARISON AND CONCLUSION

A. Networks Comparison

In terms of model size, the Hierarchical SAGPool network stands out as the most compact, occupying only 0.3 MB. On the other hand, the FeatureConcat approach exhibits a larger model size, consuming 11.418 MB. The PointNetBasedGraphPooling network follows closely, with a model size of 12.136 MB, while PointNet takes up the most space with 13.259 MB.

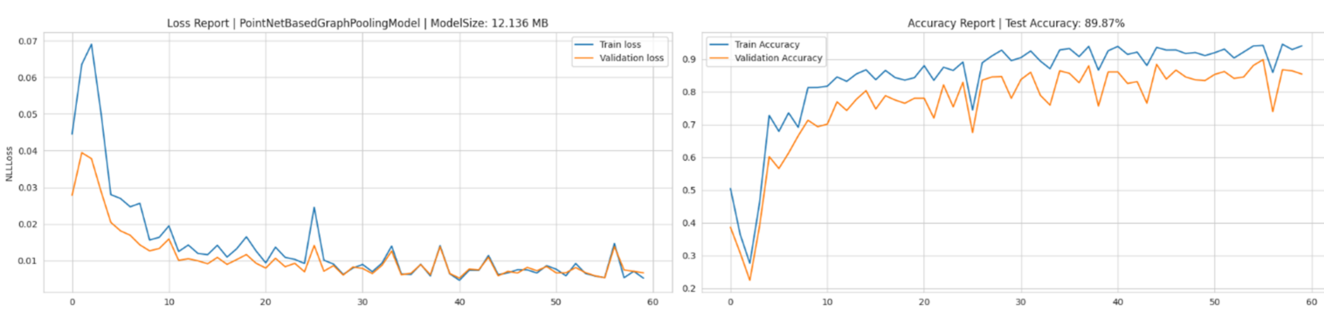

Fig. 8. Training result for graph classification based on PointNet features approach.

TABLE I

TRAINING RESULTS OF FOUR NETWORKS. THIS TABLE PROVIDES A COMPREHENSIVE OVERVIEW OF THE RESULTS OBTAINED FROM TRAINING FOUR NETWORKS, INCLUDING TRADITIONAL POINTNET, HIERARCHICAL SAGPOOL, FEATURE CONCATENATION, AND GRAPH CLASSIFICATION BASED ON POINTNET FEATURES APPROACHES. FOR EACH NETWORK, THE FEATURES USED DURING TRAINING, TEST ACCURACY ACHIEVED, MODEL SIZE, AND CORRESPONDING LEARNING RATE ARE REPORTED. THE BEST-PERFORMING RESULTS ARE DENOTED BY BOLD FONT, WHILE THE SECOND-BEST RESULTS ARE UNDERLINED, OFFERING VALUABLE INSIGHTS INTO THE PERFORMANCE OF EACH APPROACH

Network	***Features***	***Acc***	***Size***	***LR***
PointNet	3 (XYZ)	85.3	13.259 MB	0.001
Hierarchical SAGPool	3 (XYZ) + 7 (Centralities)	81.61	0.3 MB	0.01
FeatureConcat	3 (XYZ) + 7 (Centralities)	92.07	11.418 MB	0.005
PointNetBasedGraphPooling	3 (XYZ) + 7 (Centralities)	89.87	12.136 MB	0.005

When considering test accuracy, FeatureConcat outperforms other networks, achieving an impressive accuracy of 92.07

Regarding generalization, FeatureConcat showcases the best performance, successfully capturing complex patterns and dependencies in the data. PointNetBasedGraphPooling also exhibits strong generalization capabilities, owing to the fusion of local and global features. PointNet performs well in generalizing to unseen data, while Hierarchical SAGPool shows slightly lower generalization performance.

In conclusion, the FeatureConcat approach strikes a balance between model size and test accuracy, presenting a superior choice for point cloud classification tasks. PointNetBasedGraphPooling follows closely, showcasing promising results by leveraging the intersection of PointNet and SAGPool. The PointNet architecture excels in efficiency and is a viable option for point cloud classification. The hierarchical SAGPool approach offers a compact model size but falls slightly behind in accuracy and generalization. Overall, the fusion approaches demonstrate considerable potential for enhancing point cloud classification, with FeatureConcat standing out as the most robust and effective solution in this study.

B. Conclusion and review the report

The task of this research was to explore and enhance the point cloud classification using feature engineering, particularly by incorporating graph-based representations and node centralities. To achieve this, two main traditional approaches, PointNet and SAGPool, were employed as the basis for investigation. The study focused on developing and evaluating two new fusion approaches, namely FeatureConcat and PointNetBasedGraphPooling, which combined the strengths of PointNet and SAGPool architectures.

Feature engineering played a crucial role in improving the representation of point cloud data. By building graphs and utilizing node centralities, the relationships and significance of each point in the point cloud were effectively captured. The inclusion of node centralities, such as betweenness, closeness, Katz, PageRank, eigenvector, harmonic, and load centralities, provided valuable connectivity information, aiding in more accurate and robust classification.

The traditional approaches, PointNet and SAGPool, served as fundamental baselines for the new fusion approaches. PointNet, being a pioneering architecture in point cloud classification, showcased efficiency in handling unordered data. SAGPool, on the other hand, demonstrated the capability of adaptive feature pooling using graph structures, proving beneficial in capturing global context.

The newly proposed approaches, FeatureConcat and PointNetBasedGraphPooling, exhibited significant improvements in point cloud classification. FeatureConcat effectively combined high-level features from PointNet and SAGPool, leading to remarkable accuracy gains. PointNetBasedGraphPooling leveraged the intersection of PointNet and SAGPool features, further enhancing classification performance.

The results indicated that the fusion of SAGPool and PointNet significantly improved test accuracy compared to the individual approaches. FeatureConcat achieved an impressive test accuracy of 92.07

In conclusion, this research successfully explored and enhanced point cloud classification through feature engineering. The incorporation of graph-based representations and node centralities proved instrumental in achieving more accurate and efficient classification. The newly proposed fusion approaches, FeatureConcat and PointNetBasedGraphPooling, demonstrated remarkable performance gains, showcasing their potential in advancing point cloud classification tasks. This study contributes valuable insights and practical approaches for handling complex point cloud data and lays the foundation for further research in this field.

C. Future works

- Diverse Dataset Exploration: To further validate the robustness and generalization of the proposed fusion approaches, it is essential to explore and evaluate their performance on different datasets with varying complexities and sizes. Investigating datasets that encompass a wider range of 3D objects and scenes will provide deeper insights into the approaches' adaptability and effectiveness in real-world scenarios.
- Extended Training Analysis: In future research, the effect of extended training periods should be explored to understand how longer training durations impact the convergence and overall performance of the fusion approaches. Additionally, investigating the influence of different batch sizes and learning rates on training dynamics and model performance will help in optimizing the training process and achieving higher accuracy.

- Fine-tuning of Fusion Approach Components: The proposed fusion approaches, FeatureConcat and PointNet-BasedGraphPooling, utilize MLP and 1D convolution layers, dropout, and batch normalization. Fine-tuning these components and hyperparameters can further enhance the network's performance and efficiency. Optimizing these details will lead to improved feature extraction and pooling, contributing to better classification results.
- Ablation Study of Used Features: An ablation study, focusing on the impact of the individual features used in the fusion approaches, would shed light on their individual importance and contribution to the classification performance. Understanding the significance of each feature can help in designing more effective and efficient feature representations for point cloud classification tasks. The source code for the proposed fusion approaches, along with the experimental setup and dataset preparation, is available at the following link: https://github.com/MohsenEbadpour/PointNet-meets-Self-Attention-Graph-Pooling-A-Synergistic-Approach-to-Point-Cloud-Classification. Researchers are encouraged to contribute to the development and completion of this work. Collaborating on refining the fusion approaches, exploring additional datasets, and conducting in-depth experiments will enhance the credibility and significance of the findings.
  Furthermore, we extend an invitation to fellow researchers to join efforts in preparing and submitting this research to a reputable academic conference or journal. By collaborating and sharing insights, we can collectively advance the field of point cloud classification and foster meaningful discussions within the scientific community.

D. Figures and Tables

a) Positioning Figures and Tables: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation "Fig. **??**", even at the beginning of a sentence.

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader. As an example, write the quantity "Magnetization", or "Magnetization, M", not just "M". If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write "Magnetization (A/m)" or "Magnetization {A[m(1)]}", not just "A/m". Do not label axes with a ratio of quantities and units. For example, write "Temperature (K)", not "Temperature/K".

Acknowledgment

The successful completion of this research and investigation is acknowledged as Mohsen Ebadpour's final project in the 3D Vision course during his Master of Science degree at Amirkabir University of Technology. Gratitude is extended to Dr. Javanmardi, the course lecturer, for providing invaluable guidance and mentorship, which played a crucial role in shaping the direction and outcomes of this academic journey. His support and expertise were highly appreciated throughout the study, serving as a guiding force in the research process.

References

Please number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use "Ref. [3]" or "reference [3]" except at the beginning of a sentence: "Reference [3] was the first . . ."

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors' names; do not use "et al.". Papers that have not been published, even if they have been submitted for publication, should be cited as "unpublished" [4]. Papers that have been accepted for publication should be cited as "in press" [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

References

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.
[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
[3] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
[4] K. Elissa, "Title of paper if known," unpublished.
[5] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.
[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.