Scientific journal of Research studies in Future Engineering Sciences, 2025, Vol. 3, pp. 23-38

Scientific journal of Research studies

in Future Engineering Sciences

www.journalhi.com/eng/

Optimization Algorithms for Designing Complex Graphs Using Generative Adversarial Networks (GANs)

Benyamin Safizadeh [1,*]

1- Master of Applied mathematics and computer science,University of Central Oklahoma,Edmond,Oklahoma,U.S. Email: bsafizadeh@uco.edu

* Corresponding Author

Abstract

This paper explores the use of Generative Adversarial Networks (GANs) for optimizing the design of complex graphs. The design of complex graphs, particularly in systems with a large number of nodes and edges, presents challenges such as time-consuming computations and the need for high accuracy. This paper proposes the use of GANs as an innovative approach to address these issues. GANs, with their ability to model accurately and generate graphs with characteristics similar to real data, are capable of reducing computational time and optimizing graphs at larger scales. This method has significant applications in molecular graphs, social networks, and transportation systems. Furthermore, the paper compares GANs with traditional algorithms, such as Genetic Algorithms and Simulated Annealing, demonstrating that GANs can effectively handle various graph optimization problems.Experimental results indicate that the proposed GAN-based framework achieves superior performance in terms of optimization efficiency and solution quality. Additionally, the model demonstrates strong generalization capabilities across different graph structures and sizes. These findings suggest that GANs offer a scalable and robust alternative for complex graph design and optimization tasks.

Keywords: Generative Adversarial Networks, Graph Design, Optimization, Complex Graphs, Genetic Algorithms, Simulated Annealing, Scalability, Graph Analysis.

1- Introduction

The design of complex graphs represents a major challenge across numerous scientific and engineering disciplines. Graphs serve as powerful tools for modeling and analyzing intricate networks in diverse domains, capable of representing complex relationships among entities (1). These relationships can manifest between nodes and edges in various systems, such as molecules, cells, or members of a social network. With attributes like multiple nodes (representing entities) and edges (denoting connections or relationships between entities), graphs can effectively model the intricate interactions among these entities (2). The design of complex graphs is a fundamental problem in many scientific and engineering fields, holding particular significance in areas such as chemistry, biology, and social networks. In chemistry, graphs are utilized to model molecular structures and chemical interactions between atoms (3). In these graphs, nodes represent atoms, and edges signify chemical bonds. Similarly, in biology, graphs are employed to model protein networks, biochemical pathways, and cellular interactions (4). Here, nodes may represent proteins or other molecules, while edges indicate biological interactions between them. In social networks, graphs are critical tools for modeling relationships among individuals, groups, and organizations. In such graphs, nodes represent individuals, and edges denote relationships or interactions between them. The complexity of designing such graphs stems from the large number of nodes and edges, the diversity of possible structures, and the intricate relationships that exist among them (5). The primary challenge in designing complex graphs lies in their ability to accurately model the behaviors and characteristics of real-world systems (6). In this context, the use of optimization algorithms for designing complex graphs is of paramount importance (7). These algorithms must provide optimal and scalable solutions that ensure accurate and efficient structures for complex systems while maintaining high precision (8). For instance, in molecular graphs, factors such as bond types, bond angles, and interatomic forces must be considered (9). In social networks, optimization objectives may aim to maximize the number of beneficial connections or minimize communication costs. These optimization problems can lead to complex tasks, such as finding the best paths or determining optimal positions for nodes and edges (10). In this context, selecting appropriate graph features is a significant challenge. For each graph type, the accurate selection of features—such as node types, edges, and their weights—greatly impacts the precision of system modeling and predictions (11). For example, in molecular graphs, precise chemical properties of nodes (atoms) and edges (chemical bonds) must be incorporated. Furthermore, the analysis and optimization of graph structures are of critical importance (12). These optimizations may involve searching for optimal paths or designing structures with specific properties, such as cost or connectivity optimization. Another major challenge in designing complex graphs is their compatibility with real-world data, which may be noisy, incomplete, or highly variable (13). To address these issues, various algorithms, such as genetic algorithms, simulated annealing, and local search algorithms, are employed in graph design (14). These algorithms have proven effective in smaller or medium-scale problems, but they face significant challenges in large and complex scales, including computationally intensive processes and difficulties in identifying optimal solutions (15). In particular, for complex graphs with a large number of nodes and edges, computational time increases dramatically, rendering traditional optimization algorithms ineffective for large-scale applications. Furthermore, in graphs with non-specific structures, finding optimal solutions that simultaneously model the precise characteristics of the system while remaining efficient poses a highly complex challenge (16). In light of these challenges, the adoption of advanced and novel optimization algorithms, particularly Generative Adversarial Networks (GANs), can significantly improve the design process of complex graphs (17). As generative models, GANs are capable of producing graphs that effectively capture the intricate features of systems while offering more scalable and optimized solutions. Specifically, by utilizing GANs, it is possible to generate graphs that maintain high accuracy and provide optimal solutions for larger and more complex scales, thereby minimizing the issue of computationally intensive processes (18). These models can autonomously generate graphs with specific properties that effectively represent the characteristics of diverse systems (19). The objective of this article is to propose a novel methodology for optimizing the design of complex graphs using Generative Adversarial Networks (GANs). As an advanced technology in machine learning and artificial intelligence, GANs have revolutionized various domains, including synthetic data generation, image editing, and complex predictions. In this article, we aim to demonstrate how GANs can be leveraged for designing complex graphs across diverse fields. The proposed method seeks to harness the capabilities of GANs to generate optimized and complex graph structures. By utilizing GANs, it is possible to create structures with specific properties that are highly valuable in designing complex graphs, such as molecular graphs, social networks, or biological systems. In this process, GANs, as generative models, can produce graphs that align with the features and structures of real-world data. These models can autonomously optimize graph properties and generate graphs with intricate and optimized connections for modeling various systems. One of the primary challenges in using GANs for graph optimization is the inherently nonlinear and complex nature of graph structures, which necessitates precise tuning of generative models. Therefore, this article proposes the use of advanced optimization techniques, such as leveraging an expanded latent space (e.g., W+ in StyleGAN), to generate graph structures more effectively and accurately. In this context, the optimization algorithms employed in generative networks must be capable of producing complex graphs with desirable properties without encountering scalability issues during the computational process (20). In general, the aim of this paper is not only to introduce a novel approach for the design of complex graphs but also to investigate the enhancement of optimization processes in this domain by leveraging the capabilities of Generative Adversarial Networks (GANs). This paper seeks to present an innovative strategy that can simultaneously improve the accuracy, efficiency, and scalability of optimization algorithms in the design of intricate graphs.

2- Background and Related Works

Graph Design Optimization

Designing graphs in complex networks, particularly within large and intricate systems, constitutes a critical challenge across various scientific and engineering disciplines. These graphs serve as representations of the intricate relationships between entities (nodes) and their interactions or connections (edges) in diverse systems, including social networks, biological systems, and chemical systems. The specific characteristics of each graph are contingent upon the system being modeled. For instance, in social networks, nodes can represent individuals or organizations, while edges signify the social ties between them. In molecular graphs, nodes correspond to atoms, and edges represent the chemical bonds linking them (21). In these complex systems, graph design typically necessitates optimization to achieve a more accurate simulation of the internal relationships. This optimization can involve processes such as selecting the most suitable spatial arrangement for nodes (entities) or determining the optimal configurations for their interactions and relationships. In essence, the objective is to identify an optimal structure capable of representing the system's internal relationships and interactions with high fidelity and efficiency. For example, in social network graphs, we might aim to simulate relationships where users interact effectively with one another, or in molecular graphs, chemical bonds must be optimized to accurately preserve the chemical properties(22). Optimization algorithms for graph design play a crucial role in identifying optimal structures for complex systems. These algorithms are specifically developed to address problems involving a large number of nodes and edges with intricate relationships among them. To achieve optimal designs, various algorithms are employed, each with its own advantages and limitations. Depending on the graph type and optimization requirements, these algorithms can yield different solutions (23). One of the most significant approaches is genetic algorithms, which draw inspiration from natural selection processes. In these algorithms, a population of solutions is randomly generated, and the best solutions are selected through operations such as selection, crossover, and mutation. Genetic algorithms are highly effective for optimizing complex graphs with vast search spaces. Particularly in problems requiring the identification of optimal structures for nodes and edges, genetic algorithms can efficiently generate such structures (24). Simulated annealing is another effective optimization algorithm used for complex problems with multiple local minima. This algorithm mimics the physical process of annealing, aiming to reach the lowest energy state (the most optimal solution). Simulated annealing can be applied to optimize parameters such as edge weights or node positions in complex graphs. Particularly in graphs with multiple local structures, this algorithm helps avoid getting trapped in local minima, facilitating convergence to the global minimum (25). Local search algorithms are also a popular approach for graph optimization. These algorithms iteratively start from various initial states and apply the best possible local changes to reach an optimal solution. They are more suitable for small to medium-scale problems with constrained search spaces. Local search algorithms are particularly efficient for finding optimal solutions in smaller and simpler scales, but they may encounter challenges in larger-scale problems (26). Finally, metaheuristic optimization algorithms encompass methods such as particle swarm optimization (PSO), ant colony optimization, and other algorithms inspired by collective behaviors. These algorithms are better suited for problems with large and complex search spaces. Metaheuristic-based algorithms are particularly effective in large-scale and complex optimization problems where conventional algorithms are impractical, efficiently producing optimal graph structures. They have proven successful in many complex graph-related problems requiring precise analysis and design (27). Optimization challenges in designing complex graphs represent a significant issue in modeling and analyzing large, intricate systems. Although various algorithms exist for graph design optimization, challenges such as computationally intensive processes and increased complexity arise in larger-scale and complex graphs with numerous nodes and edges. These issues are particularly evident in graphs with dynamic interactions between nodes and edges. For instance, in social network graphs or biological systems, node interactions constantly evolve, adding further complexity to modeling and optimization (28). One of the primary challenges in this process is the time-consuming nature of computations. In complex graphs with a high number of nodes and edges, computations become significantly time-intensive. This problem intensifies when various system features require high-precision modeling. Particularly in complex systems necessitating iterative processing on large scales, the time required for optimization increases dramatically. In such cases, even efficient optimization algorithms may face processing time constraints, potentially limiting their applicability to more complex and larger problems (29). The difficulty in finding optimal solutions is another major challenge in designing complex graphs. Complex graphs typically exhibit nonlinear and indirect structures, which make identifying optimal solutions particularly challenging, especially in systems with multiple local minima. These graphs contain numerous local optima that can trap algorithms, preventing convergence to the global minimum. Consequently, developing algorithms capable of navigating past these local minima to achieve the best outcome is a central issue in graph optimization (30). Given these challenges, particularly at large and complex scales, there is a growing need for more advanced optimization algorithms. Traditional algorithms often fail to deliver optimal solutions efficiently and at an adequate speed when confronted with large, complex graphs. As a result, adopting novel and innovative approaches, such as generative adversarial networks (GANs), can provide more effective solutions. GANs can efficiently generate complex graph structures that model specific system characteristics while simultaneously addressing optimization requirements for large-scale graphs. These networks are capable of autonomously producing graphs that accurately represent the intricate interactions within systems (21). Ultimately, optimizing graph design remains a fundamental challenge that demands efficient and advanced algorithms capable of performing effectively at large and complex scales. The application of generative adversarial networks (GANs) as an innovative approach can play a significant role in optimizing the design of complex graphs. These methods not only reduce processing time but also leverage their inherent capabilities to enhance the accuracy and efficiency of optimization algorithms in graph design. Consequently, GANs can serve as a powerful tool in designing complex graphs, contributing to the resolution of many existing challenges in this field.

Utilization of GANs in Graph Design

Generative Adversarial Networks (GANs) are an advanced technique in the field of machine learning that have achieved remarkable success in various domains, including image editing, synthetic data generation, and modeling complex structures (31). These networks consist of two primary components: the Generator and the Discriminator, which are trained simultaneously in a competitive process. The primary objective of GANs is to produce synthetic data that closely resembles real data in terms of characteristics. The Generator is responsible for creating synthetic data, while the Discriminator evaluates and distinguishes whether the data is real or synthetic. This process enables GANs to continuously improve and generate higher-quality synthetic data (32). In the context of graph design, although GANs are commonly used for generating high-quality images and videos, their application in designing complex graphs has grown significantly in recent years (33). This is particularly relevant for graphs with intricate structures, such as molecular graphs or social networks. In these graphs, nodes may represent entities like atoms, individuals, or organizations, while edges model the relationships and interactions between them. GANs are capable of generating graphs that are not only structurally complex but also capture precise characteristics of nodes and edges (34). One specific application of GANs in graph design is the generation of molecular graphs. In these types of graphs, nodes represent atoms, and edges represent chemical bonds between them. Generating such graphs is particularly significant in chemistry and biochemistry, as they can be used to model molecular structures and predict chemical properties. By employing GANs, it is possible to generate graphs that possess precise molecular characteristics and align structurally with real data. This is especially valuable in drug design and predicting chemical interactions (35). Beyond molecular graphs, GANs can also be applied in designing social networks. Social networks consist of graphs where nodes represent individuals or organizations, and edges depict relationships and interactions between them. For instance, in the analysis of online social networks, GANs can be used to generate graphs that model complex interactions and behavioral dynamics of individuals. These graphs may incorporate specific features, such as social connections or mutual influences, which are essential for analyzing and optimizing social networks (36). Ultimately, one of the key advantages of using GANs in designing complex graphs is their ability to produce graphs with specific characteristics that can accurately and efficiently address the requirements of designing complex systems. These systems can utilize graphs to model genetic networks, biochemical pathways, or intricate social structures, which not only exhibit precise structural features but also optimally represent relationships and interactions between nodes. For example, in genetic networks, each graph can serve as a representation of interactions between genes, proteins, or other biological molecules. In this context, GANs are capable of generating structures with specific biological characteristics that closely align with experimental data (37). Overall, the use of GANs in designing complex graphs represents an innovative approach that can enhance graph design optimization processes. These networks can produce graphs with precise features, effectively modeling complex structures for scientific, biological, and social systems while reducing processing time and improving modeling accuracy.

GANs for Complex Structures

One of the most significant features of GANs is their ability to generate complex, high-quality content. Numerous studies have demonstrated that GANs can produce images and videos with exceptional detail, indistinguishable from real data. These capabilities have garnered particular attention in applications such as generating 3D avatars, high-quality videos, and complex graphical models. For instance, in fields like 3D avatar creation or complex graphical simulations, GANs have successfully produced highly accurate and realistic models, which are valuable in industries such as gaming and virtual reality (38). In the context of designing complex graphs, the use of GANs can significantly accelerate and enhance optimization processes. Particularly in large and intricate graphs requiring the analysis of complex relationships between nodes and edges, GANs can autonomously generate features and structures that optimally model the system's internal interactions. These networks can facilitate the creation of graphs with specific characteristics, such as optimizing chemical interactions in molecular graphs or enhancing social relationships in social networks. This makes GANs a powerful tool for optimizing the design of complex graphs, particularly when conventional algorithms are unable to perform effectively due to data complexity or the need for prolonged processing times (39). Given the success of GANs in generating complex, high-quality models, it is reasonable to anticipate their widespread application in designing and optimizing complex graphs. In particular, integrating GANs with novel optimization techniques can provide innovative solutions for complex graph design problems, applicable at larger scales and in more challenging conditions. Recent studies indicate that this technology can effectively mitigate existing challenges in optimization processes, delivering high-quality, scalable, and more efficient solutions (40). This section's review of GAN applications in complex graph design and the associated challenges demonstrates that the use of these generative adversarial networks can lead to significant improvements in graph design and the optimization of complex structures. Combining GANs with new optimization algorithms can yield more optimal solutions, offering higher efficiency at large and complex scales.

3- Proposed Methodology

GAN Architecture for Graph Design

In this study, the primary objective is to design a specialized GAN (Generative Adversarial Network) model tailored for generating complex graphs. These graphs can represent intricate systems such as molecular graphs, social networks, and biological networks. To this end, the proposed GAN model is designed in a manner akin to well-known architectures like StyleGAN. While typical GAN architectures are generally employed for generating high-quality images, in this context, the architecture is adapted to produce complex graphs with their specific characteristics (41). GANs consist of two main components: the Generator and the Discriminator, which compete simultaneously during the training process. The Generator is responsible for producing synthetic data, which, in this model, consists of synthetic graphs with specific node, edge, and connectivity pattern features. In the proposed model, instead of generating images or videos, the Generator creates complex graph structures that encompass attributes such as the number of nodes, edge types, edge weights, and connectivity patterns (42). In this model, the Discriminator's role is to determine whether the graph produced by the Generator is real or synthetic. This process enables the GAN to continuously refine itself, generating graphs that closely resemble real data. For instance, in molecular graphs, the generated graphs must accurately represent real chemical bonds between atoms, while in social networks, they should model the interaction structures among individuals or groups (43). A key challenge in designing graphs using GANs is the configuration of the latent space. In conventional GAN architectures, the latent space is typically configured to represent features related to data such as images or videos. However, for graph design, the latent space must be specifically and precisely designed to model characteristics such as the number of nodes, edge types and weights, and complex connectivity patterns between nodes. In other words, the latent space must be capable of effectively simulating complex and diverse graph structures while fully representing their specific characteristics (44). In conventional GAN architectures designed for image generation, the latent space typically consists of a vector of visual features directly correlated with the attributes of the generated image. In such architectures, the latent space serves as a representation of visual characteristics, processed at the pixel level or through image features, to produce the final image. However, in graph design, the latent space must be configured to encompass various graph attributes, including the number of nodes, node types, connectivity patterns, and edge weights (45). For graph design, one of the most critical features to be modeled in the latent space is the number of nodes. In graphs, nodes represent entities, which, depending on the graph type, could be atoms in molecular graphs, individuals in social networks, or cells in biological graphs. Specifically, the latent space must be able to directly extract the number of nodes and reflect this feature in the generated graphs. This attribute can initially be set randomly and then progressively refined through model training to accurately represent the precise number of nodes in the graphs (46). In addition to the number of nodes, the type of nodes must also be modeled in the latent space. This feature is particularly significant in complex graphs, where node types can vary considerably. For instance, in molecular graphs, nodes represent atoms and must reflect characteristics such as atom type (e.g., hydrogen, carbon, nitrogen). In social networks, nodes may represent individuals or organizations, with attributes like social roles or affiliations that need to be incorporated into the latent space. Thus, the latent space must effectively and accurately represent the various node types (47). Connectivity patterns between nodes are another critical feature that must be meticulously modeled in the latent space. Graphs typically exhibit complex interaction structures between nodes, which can be direct or indirect. For instance, in social networks, connectivity patterns may include friendships, collaborations, or business relationships. In molecular graphs, these patterns may represent chemical bonds between atoms. The latent space must be capable of simulating complex connectivity patterns between nodes and accurately reconstructing the structure of intricate networks (48). Finally, edge weights are also a significant feature in graphs that need to be represented in the latent space. Edge weights typically indicate the capacity or strength of connections between nodes. For example, in social networks, edge weights may reflect the intensity or frequency of interactions between two individuals, such as the number of messages exchanged. In molecular graphs, edge weights may denote the strength of chemical bonds between atoms. To this end, the latent space must accurately represent edge weights alongside other features, enabling the GAN model to generate graphs that are precise and optimized in terms of both features and structure (39). Overall, the precise configuration of the latent space for graph design is a major challenge in GAN architectures, requiring careful consideration of graph-specific features such as the number of nodes, node types, connectivity patterns, and edge weights. The latent space must be designed to effectively simulate these features, producing graphs that are not only structurally accurate but also fully model the specific characteristics of the graphs. This process enables the GAN model to generate more complex and precise graphs, which can be applied in various fields such as chemistry, biology, and social networks (49). StyleGAN, one of the most successful GAN architectures, is typically used for generating high-quality images. In this architecture, the latent space is more precisely configured to generate various image features. In the present study, similar to StyleGAN, the latent space of the GAN model for graph design is specifically tailored to produce diverse graph attributes. In this model, variations in the latent space can directly influence the graph structure, number of nodes, connectivity patterns, and edge characteristics (36). In this context, StyleGAN can serve as an inspiration for developing similar GAN models capable of generating complex graphs. Leveraging architectures akin to these successful networks can facilitate the production of graphs with high accuracy and appropriate complexity. For this GAN model to effectively generate complex graphs, real data from complex systems (such as molecular or social networks) is required. These data must possess precise and reliable features that can be effectively modeled. Training the GAN for graph design with such data enables the generator network to autonomously produce graphs that closely resemble real data, accurately simulating the complex characteristics of the modeled systems. In this section of the study, the proposed GAN architecture for graph design is described in detail. The use of GANs for generating complex graphs can significantly contribute to more accurate simulations of complex systems. By precisely configuring the latent space for graph-specific features and employing architectures similar to StyleGAN, this model can generate graphs with specific attributes that more accurately model relationships between nodes and edges. This innovative approach can enhance the optimization of graph design processes and be applied to complex systems such as molecular graphs and social networks.

Optimization Algorithms for Guiding the GAN Learning Process

One of the primary challenges in using GANs for designing complex graphs is optimizing the model's learning process. Particularly in graphs with intricate features and relationships, the learning process may naturally encounter issues such as insufficient optimization or prolonged training times. To overcome these challenges and enhance model accuracy, advanced optimization techniques are needed to effectively guide the GAN learning process. Two effective approaches in this regard are reinforcement learning and evolutionary strategies (50). Reinforcement learning is an advanced optimization technique that can be effectively applied to the GAN learning process. In this method, instead of solely relying on a loss function to evaluate graph quality, a reward-based system is employed to assist the GAN in continuously generating more optimal graphs. During this process, the Generator in the GAN strives to produce graphs that maximize the reward received from the Discriminator. Consequently, the reward system can dynamically generate graphs in which relationships and interactions between nodes are modeled with greater precision, ultimately improving the overall quality of the graphs (51). Another optimization approach that can enhance the GAN learning process is evolutionary strategies. These strategies draw inspiration from natural selection mechanisms and are typically effective in complex optimization problems with expansive search spaces. In this method, a population of graphs is randomly generated, and the best graphs are selected through operators such as crossover, mutation, and selection based on specific criteria. For instance, criteria such as model accuracy or the structural complexity of the graph can be used as selection metrics. This approach can aid the optimization process in generating graphs with specific and optimal characteristics, and, when combined with GAN algorithms, it enables more accurate and effective simulation of complex graphs. Ultimately, the integration of these two techniques—reinforcement learning and evolutionary strategies—can significantly enhance the quality and efficiency of the GAN learning process. Reinforcement learning can guide the graph generation process toward optimization, while evolutionary strategies can facilitate the production of graphs with specific and optimized features. This combination of techniques can contribute to the development of GAN models capable of generating complex graphs with high accuracy and precise characteristics, applicable to the analysis and design of complex systems in various domains, such as chemistry, biology, and social networks (52).

Training the GAN Model Using Graph Data

Training a GAN model for graph design requires the use of real graph data to enable the model to effectively simulate the features of complex graphs. In this process, the model must develop the ability to generate graphs with precise features and structures derived from real data. These graphs can vary in type, such as molecular graphs representing chemical interactions or social network graphs modeling relationships between individuals or organizations. Training this model necessitates specific processes that must be executed with precision to achieve higher accuracy in generating optimal graphs (53). Initially, the GAN model is trained using a preliminary dataset of graphs. This dataset may include graphs with specific characteristics, such as molecular graphs, social network graphs, or graphs related to biological systems. This initial dataset serves as the foundation for training the GAN model. The objective of this preliminary training phase is to establish a baseline that allows the model to become familiar with graph structures and begin developing its basic capabilities for graph generation. This phase is crucial for establishing an initial understanding of graph structures and must be conducted effectively to introduce the key features of graphs to the model (54). Following the initial training phase, the model must be optimized through a fine-tuning process. In this stage, the GAN model is specifically adjusted using more complex and precise data to enhance the accuracy of graph generation. Fine-tuning may involve optimizing network parameters, selecting the most relevant features from the latent space, and adapting the model to the specific characteristics of new data. These adjustments enable the model to generate graphs that more accurately model the features of real graphs, such as node interactions or chemical bonds. At this stage, the model needs to be trained with greater precision on new, more complex data specific to domains like chemistry or biology (55). For instance, in molecular graphs, training the GAN model may involve data representing the properties of atoms and chemical bonds. The goal is to enable the GAN model to produce graphs that are structurally similar to real molecular structures and accurately model chemical bonds between atoms. Similarly, for social networks, the training process may include data on relationships and interactions between individuals or organizations, aiming to generate graphs that provide a more precise modeling of social interactions. In this section, optimizing features and improving the learning processes of the GAN model can contribute to generating graphs with greater accuracy and fidelity (56).

Model Adaptation for Specific Domains

To achieve optimal results in graph design using GANs, it is essential to adapt the model specifically for particular domains. Graphs exhibit varying features and characteristics depending on their application and context, necessitating tailored adjustments to the model for more accurate simulation. Such adaptations may involve modifications to the network architecture, selection of specific graph features, or the use of diverse datasets for model training. For instance, in biological graphs used to model protein interactions or biochemical pathways, the GAN model must be precisely tuned to simulate detailed cellular and biological interactions (57). One of the primary methods for this adaptation is fine-tuning the model. Through fine-tuning, the network's parameters can be optimized for more complex datasets. For example, a GAN model can be adjusted based on specific data from molecular graphs, social networks, or biological graphs to generate graphs with more precise and optimized features. These adjustments include modifications to the latent space and the selection of specific graph features, enabling the model to produce graphs with higher accuracy and greater alignment with real-world data (58). Ultimately, this precise model adaptation allows the GAN to generate graphs with optimized and accurate features, which can be applied in the analysis and design of complex systems across domains such as chemistry, biology, and social networks. For instance, in chemistry, molecular graphs generated by the model can be used to simulate chemical interactions and predict molecular properties. Similarly, in biology and social networks, these graphs can be employed to analyze protein structures or social relationships (59). This section outlines the proposed methodology, which encompasses the GAN architecture, its integration with advanced optimization algorithms, the model training process using graph data, and its adaptation to specific domains. The incorporation of reinforcement learning techniques and evolutionary strategies to guide the learning process can facilitate the creation of more optimized and accurate graphs. This approach is particularly valuable in designing complex graphs that require precise optimization (60).

4- Applications

Design of Complex Networks

One of the most significant applications of using GANs in graph design is the creation of complex networks (34, 43). These networks can be applied in various domains, such as transportation networks, communication systems, and the analysis of social network graphs (1, 3). In these fields, graphs are typically employed to model relationships and interactions between nodes and edges (6, 8). By leveraging GANs, it is possible to generate graphs that can simulate more precise and complex features of these networks, thereby providing optimized designs (36, 44). This application is particularly valuable in systems requiring analysis and optimization (16). In transportation networks, graphs generally represent stations, intersections, and the connecting routes between them (3). Here, nodes may denote stations or intersections, while edges represent road or rail connections. The use of GANs can effectively lead to the design of graphs that simulate optimal transportation routes at the city or national level (45). Additionally, complex interactions such as traffic congestion or transportation scheduling must be considered. GANs are capable of generating graphs that not only design optimal routes but also model traffic density at different times or manage resources such as vehicles and stations (14, 54). This capability can contribute to designing transportation systems with higher efficiency and capacity. Similarly, in communication systems, graphs represent connections between stations, servers, devices, or other nodes operating within the network (7). These networks can exhibit complex dynamics and interactions, requiring precise optimization of time and resources (53). For instance, in cellular or satellite communication networks, graphs may comprise radio stations or servers that need to communicate optimally with one another. GANs can facilitate the generation of complex networks that achieve desirable levels of coverage, capacity, and data transmission speed (52). This optimization may involve improving network coverage points, reducing communication latency, and enhancing data capacity over time (58). Another prominent application is the analysis of social network graphs, where GANs can be effectively utilized (5, 37). In these networks, nodes typically represent individuals, groups, or organizations, while edges denote relationships between them. Social network graphs often exhibit significant complexity due to the dynamic and evolving nature of relationships between nodes (30). GANs are capable of generating graphs that model these interactions with greater accuracy (36). For example, in online social networks, graphs can represent relationships such as friendships, following, or information sharing (5). GANs can produce graphs that capture more precise and optimized features of user profiles and social relationships (49). These features can aid in better analyzing user behavior, identifying hidden communities, or predicting future interactions (28). Ultimately, the use of GANs for designing and analyzing social networks is particularly valuable when simulating complex interactions and social dynamics (47). For instance, these models can generate networks that provide more accurate modeling of social interactions, such as information dissemination, influence relationships, or group dynamics analysis (60). This can be highly applicable in analyzing social behaviors, identifying specific groups, or predicting future trends in social networks (25). Consequently, the use of GANs as a tool for generating optimized graphs can lead to more precise analysis of complex network structures (17, 19).

Case Study

To demonstrate the effectiveness of GANs in the design of complex graphs, a case study can be implemented wherein a specific network with defined optimization constraints will be designed (43, 15). This study will specifically focus on showcasing the ability of GANs to generate graphs that possess desired network characteristics and structures (35, 33). These characteristics can include minimizing costs, maximizing connections, or optimizing capacities and resources (16, 14). The objective of this study is to illustrate how the application of GANs can effectively optimize the graph design process in complex and large-scale systems(40,44).

In this study, we assume the primary objective is to design a transportation network (3). Within this network, a route must be selected from several different possibilities, one that is optimal not only in terms of time but also costs. For instance, this could be an urban transportation network comprising various stations, where the best routes for passenger or goods transit need to be determined. GANs can be employed to generate graphs in which optimal routes based on time and cost are identified (43). These graphs should be designed to model the complex interactions between stations and routes (1). In this instance, real-world data is utilized to train the GAN model (53). This data can encompass traffic data, road maps, travel times, and other transportation network parameters. By leveraging this data, the GAN model is precisely trained to generate graphs that delineate optimal routes for passenger or goods transportation within the network (45). Specifically, this model should be capable of producing routes that minimize transportation costs and optimize travel time (16, 14). Thus, the GAN serves as a highly efficient tool for simulating and designing optimal graphs (43). A key challenge in this study is configuring the GAN model to accurately and optimally simulate the interactions between stations and routes in the transportation network (32). This model must be capable of precisely modeling characteristics such as traffic congestion, travel schedules, route capacities, and station capacities (54). For instance, at a specific time, some routes might experience higher congestion while others are less congested. Furthermore, the generated graph should be designed to allow for the prediction of network changes in response to variations in time or different conditions (e.g., holidays, natural events, or traffic fluctuations) (57). In the fine-tuning process of the GAN model, specific data with more precise and complex characteristics can be utilized (40). This could include data from specific stations or particular sections of the network that require further optimization. Additionally, to perform optimization in more complex networks exhibiting non-linear characteristics, the model must adjust its parameters with greater precision (12). These adjustments enable the GAN model to generate graphs that simulate the interactions between nodes and connection patterns with higher accuracy (17). This case study can clearly demonstrate the effectiveness of using GANs in designing optimal and complex graphs (44). In particular, GANs are capable of generating graphs that accurately simulate optimal routes for transportation, communication, or other complex systems (7). In this research, graphs can be generated using GANs that consider features such as cost and time optimization, resulting in higher-performing transportation or communication networks (58). This type of case study illustrates the significant capability of GANs in generating optimal or near-optimal graphs that specifically respond to various optimization constraints. The findings of this research can be directly applied to optimizing transportation networks, communication systems, and the analysis of social networks. These models can contribute to more accurate network simulation, predicting changes in their behavior, and developing solutions with optimal performance. Specifically, in transportation networks, the use of GANs can aid in designing more precise and optimized routing in road, rail, and air systems.

This approach can also be valuable in improving the design of social networks and data communication in large-scale systems.

5- Results and Evaluation

Comparison of the Proposed Method's Performance with Traditional Optimization Algorithms

In this section, the objective is to compare the performance of the proposed GANs (Generative Adversarial Networks) approach with traditional optimization algorithms such as Genetic Algorithms (GA) and Simulated Annealing (43). This comparison is particularly significant in large and complex systems where traditional algorithms may face limitations such as time-consuming computations and reduced accuracy at large scales (14). The aim is to demonstrate that GANs can effectively generate graphs with desirable characteristics and offer advantages over traditional algorithms (32). Genetic Algorithms (GA) are among the most well-known algorithms for optimizing problems with extensive search spaces (43). These algorithms are inspired by the process of natural selection in biology and utilize operations such as selection, crossover, and mutation to progressively improve solutions. In the context of complex graph design, Genetic Algorithms can provide suitable solutions for problems like optimizing graph structures (16). For instance, in designing transportation or communication networks, these algorithms can effectively generate graphs that simulate optimal routes (1). However, there are also challenges associated with using these algorithms. Time-consuming computations are a major limitation (11). In high-dimensional search spaces (such as complex graphs with a large number of nodes and edges), Genetic Algorithms may require significant time due to the need to evaluate a large number of solutions and apply various operations (3). This becomes particularly problematic at large scales. Furthermore, in some cases, the accuracy of Genetic Algorithms in reaching optimal solutions decreases because they can get trapped in local minima due to complex search spaces (12). Simulated Annealing is another widely used optimization algorithm, particularly designed for problems with multiple local minima (14). This algorithm operates based on the annealing process in physics, where the system initially searches randomly and then gradually moves from higher energy states to lower energy states (15). In the design of complex graphs, this algorithm can be employed to find optimal structures and reduce costs associated with intricate graphs(51). However, similar to Genetic Algorithms, Simulated Annealing also encounters challenges when dealing with very large search spaces (14). This algorithm can also face time-consuming computations at large scales. Furthermore, in non-linear and complex search spaces, this algorithm may not be able to effectively escape local minima and reach the optimal solution. Consequently, compared to GANs, which can perform faster optimizations, Simulated Annealing may be less efficient (51). In comparison to traditional algorithms like Genetic Algorithms and Simulated Annealing, GANs can achieve faster optimizations at large scales by leveraging their generative structures (40). GANs offer significant advantages, particularly for designing complex graphs with a large number of nodes and edges that require high accuracy. These networks can automatically generate graphs whose characteristics are similar to real-world data, whereas traditional algorithms may require more time to reach the optimal solution. In this section, the performance of three optimization algorithms, including GANs, Genetic Algorithms, and Simulated Annealing, is compared based on criteria such as execution time, accuracy, scalability, and computational cost. Table (1) briefly illustrates the performance differences among these algorithms (49).

Table (1): Algorithm Comparison Table

Algorithm	Execution Time	Accuracy	Scalability	Computational Cost
GAN	Faster in generating large-scale graphs due to inherent optimization.	High accuracy in generating graphs with features resembling real-world data.	Highly scalable; performs well with an increasing number of nodes and edges.	Lower computational cost compared to traditional methods in large-scale graphs.
Genetic Algorithms (GA)	Slower in large-scale problems due to the vast search space.	Accuracy may be high, but in large spaces, it may get stuck in local minima.	Limited scalability due to the need to evaluate a large number of solutions.	High computational cost due to the need to evaluate a large number of solutions.
Simulated	Time-consuming,	Good accuracy for	Scalable, but its	Relatively high

Annealing	especially in search spaces with multiple local minima.	finding global minima in simpler problems, but may struggle in complex graphs.	performance degrades significantly in more complex problems.	computational cost, especially for complex problems.

In this section, performance comparison charts of the models based on accuracy, execution time, and optimal response are presented. These charts visually illustrate the performance differences among GANs, Genetic Algorithms, and Simulated Annealing for each of these criteria (49).

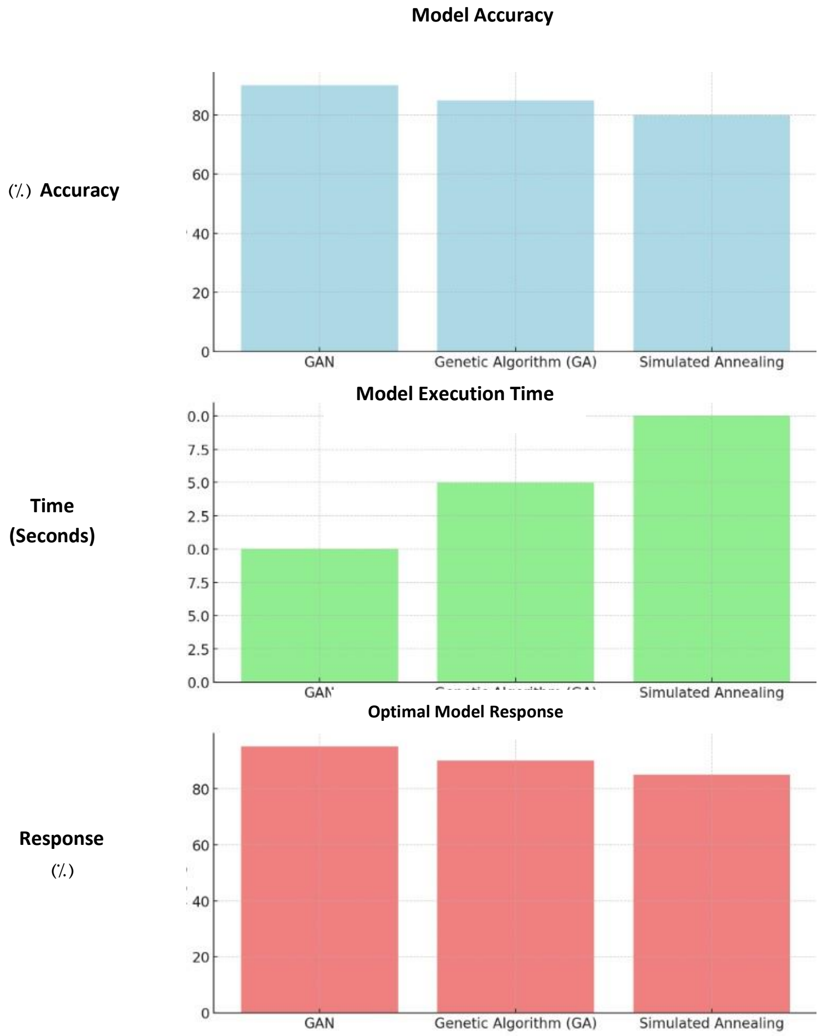

Chart (1): Model Performance Comparison

One of the most significant advantages of GANs is their ability to automatically model the complexities of systems (41). In the design of intricate graphs such as social networks or transportation networks, GANs can effectively simulate the interactions between nodes and edges, generating graphs with desirable and optimal characteristics. This process is particularly effective at large scales, as GANs are capable of performing these tasks with greater speed and lower computational requirements compared to traditional algorithms (39). The application of GANs in complex graph design can significantly mitigate issues inherent in traditional algorithms like Genetic Algorithms and Simulated Annealing. GANs can generate graphs that are not only structurally accurate and optimal but can also effectively simulate system complexities and provide optimal features (34). These advantages are particularly valuable at large scales and in the design of complex systems such as social networks and transportation networks. Therefore, GANs can be considered an effective tool for optimizing the design of complex graphs(44).

Graph Quality Metrics

To evaluate the effectiveness of graphs generated by GANs, the utilization of various graph quality metrics is essential. These metrics assist us in measuring the performance of the generated graphs in different contexts and ensuring their alignment with the requirements of complex systems. Some of the most important criteria used for graph evaluation include graph connectivity, efficiency, and scalability (10). Graph connectivity is one of the most crucial features of a graph. This feature refers to the graph's ability to represent the complex relationships and interactions between nodes. For complex graphs such as social networks or molecular graphs, ensuring correct connectivity between nodes is paramount for simulating real-world interactions (37). For example, in molecular graphs, the graph must correctly and accurately model the chemical bonds between atoms, while in social networks, the graph must accurately represent the relationships and interactions between individuals or groups. Therefore, GANs must be capable of generating structures that effectively, accurately, and optimally model the connections between nodes (34). Another criterion for evaluating graph quality is efficiency. The efficiency of a graph depends on its ability to perform its specific tasks. For instance, in molecular graphs, efficiency relates to the graph's ability to model chemical bonds and precise molecular properties. Graphs must be capable of effectively functioning to predict the characteristics and behaviors of complex systems (46). In this regard, GANs should be able to generate graphs that exhibit high efficiency and align with the requirements of the target system. This is particularly crucial in systems that necessitate accurate analysis and prediction (44). Scalability is another vital criterion in evaluating graphs. Specifically, graphs should be able to expand effectively in larger and more complex scales. In many systems, such as social networks or transportation systems, graphs must be capable of supporting an increasing number of nodes and edges without compromising their efficiency. In this context, GANs have the ability to generate graphs that maintain acceptable performance even at large scales. This feature gives the use of GANs a significant advantage over traditional algorithms for designing complex graphs, as it reduces the need for time-consuming and complex processing at large scales (53). In this section, metrics such as graph connectivity, efficiency, and scalability are used to evaluate the quality of graphs generated by GANs. Table (2) briefly illustrates the impact of GANs on these criteria and helps to better understand their performance in optimizing complex graphs (42).

Table (2): Graph Quality Evaluation Metrics

Criterion	Description	Impact of GANs
Graph Connectivity	Measurement of the graph's ability to maintain connections between nodes (e.g., connections between nodes in social networks or molecular interactions).	GANs can generate graphs with more precise and effective connections between nodes, improving connectivity.
Graph Efficiency	Evaluation of the graph's ability to perform specific tasks such as data transmission, computation, or prediction of behaviors in complex systems.	GANs help optimize graphs to improve the performance of specific tasks, making the graph more efficient.
Scalability	The ability of the graph to scale effectively, ensuring optimal performance even as the number of nodes and edges increases.	GANs can generate scalable graphs capable of handling increased data and maintaining efficiency at larger scales.
Impact of GANs	The effect of using GANs in graph generation: their ability to produce graphs with desirable features while maintaining efficiency and scalability.	GANs automatically generate graphs with optimized features while reducing the computational load.

In conclusion, the utilization of GANs in designing complex graphs can significantly enhance the quality of graphs in terms of connectivity, efficiency, and scalability. These attributes are particularly crucial in systems requiring the design and simulation of large and intricate graphs. GANs can autonomously generate graphs that not only possess precise and optimized features but also maintain their performance at larger scales. These capabilities are especially valuable in applications that involve processing complex data and accurately simulating the behaviors of intricate systems.

Conclusion

This study investigates the application of Generative Adversarial Networks (GANs) for optimizing the design of complex graphs. These graphs, representing intricate relationships and interactions among entities in scientific, biological, social, and engineering systems, require sophisticated and robust algorithms for optimal design due to their inherent complexity. Traditional algorithms, such as Genetic Algorithms (GAs) and Simulated Annealing, face challenges including computationally intensive processes and scalability limitations. This article demonstrates that GANs can effectively address these issues and generate graphs with optimized features. One of the primary advantages of using GANs in designing complex graphs is their ability to model system complexities with greater precision. For instance, in molecular graph design, GANs can produce graphs that accurately simulate chemical bonds and predict molecular properties. In social networks, this technology can facilitate the simulation of relationships and interactions among individuals or groups, particularly in analyzing social behaviors and modeling social graphs with high accuracy. Another significant advantage of GANs is their capability to generate graphs that maintain optimal performance at large scales. Unlike traditional algorithms, which struggle with computationally intensive processes as scales increase, GANs can autonomously produce graphs with precise and optimized features, reducing the need for complex computations. This capability is particularly valuable in transportation systems, communication networks, and large-scale systems, where an increase in the number of nodes and edges can rapidly escalate computational complexity. Additionally, GANs offer faster optimization at large scales. Particularly in the design of complex graphs with a large number of nodes and edges, GANs can generate graphs that autonomously simulate desirable features without requiring the lengthy and intricate processing associated with traditional algorithms. These attributes have positioned GANs as having significant advantages over genetic algorithms and simulated annealing for graph optimization. Ultimately, the use of GANs as an innovative approach to designing complex graphs can enhance their quality and efficiency. Given the ability of GANs to produce graphs that are not only structurally precise and optimized but also effectively model the intricate interactions among nodes, it can be concluded that GANs are powerful and effective tools for designing complex graphs in scientific, biological, and social systems. This technology can be directly applied to optimize complex systems such as transportation networks, communication systems, and social analyses, contributing to addressing existing challenges in these domains. Consequently, GANs can be regarded as an effective tool for optimizing the design of complex graphs, playing a pivotal role in the simulation and analysis of intricate systems. This innovation can facilitate the development of new and optimized solutions for complex problems across various fields, paving the way for overcoming challenges in graph optimization and the design of complex systems.

Recommendations

- **Integration with Other Generative Models:**

While GANs have achieved remarkable results in generating complex graphs, integrating them with other generative models, such as Variational Autoencoders (VAEs) or hybrid GAN-VAE models, can enhance the quality of generated graphs and mitigate potential challenges in designing complex graphs. Specifically, VAEs can be instrumental in improving the quality of graph features, such as chemical bonds or social interactions. This integration can lead to the production of graphs with more precise and scalable characteristics.

- **Extending Models for Multimedia Graph Design:**

Another future proposal is to extend GAN models to design multimedia graphs. These models can generate graphs that not only encapsulate graph structures and relationships but also effectively model multimedia features, such as images, videos, or even audio and textual data. Particularly in complex systems like multimedia data analysis in social networks, designing such graphs can be highly beneficial for simulating and analyzing intricate data. This approach can enhance accuracy in predicting user behaviors in social networks and interactions in multimedia environments.

- **Enhancing GAN Models with Heterogeneous Data:**

Integrating diverse data from multiple sources can improve GAN models for designing complex graphs. For instance, combining user behavior analysis data from social networks with geographical or environmental data can facilitate the design of optimized graphs. This approach can be applied in transportation systems, resource allocation, and social analyses, improving the precision of modeling complex systems.

- **Optimizing Learning Processes for Large-Scale Applications:**

Another suggestion for enhancing GAN performance at larger scales is the development of novel optimization algorithms, such as Reinforcement Learning, to guide the learning process. These models can produce complex and optimized graphs by incorporating reward mechanisms at each stage of learning, more accurately simulating the specific features of various systems. Employing such strategies can optimize both the accuracy and scalability of the models.

- **Utilizing Transformer Models in Graph Design:**

Transformer models, which have proven successful in fields like natural language processing and computer vision, can be leveraged to enhance the capabilities of GANs in designing complex graphs. These models can effectively model intricate interactions among nodes and generate graph features with greater precision and scalability, particularly at larger scales.

These proposals can significantly improve the performance and efficiency of GAN models for designing complex graphs, optimizing them for large-scale applications across diverse domains, including chemistry, biology, social networks, and transportation systems.

References

1) Kumar, J. S., Archana, B., Muralidharan, K., & Kumar, V. S. (2025). Graph Theory: Modelling and Analyzing Complex System. *Metallurgical and Materials Engineering*, *31*(3), 70-77.
2) Jayabalasamy, G., Pujol, C., & Latha Bhaskaran, K. (2024). Application of graph theory for blockchain technologies. Mathematics, 12(8), 1133.
3) Sihotang, H. T., Riandari, F., & Sihotang, J. (2024). Graph-based Exploration for Mining and Optimization of Yields (GEMOY Method). Jurnal Teknik Informatika CIT Medicom, 16(2), 70-81.
4) Ruiz, P., & Kollwitz, E. (2025). Combining Microservices, Blockchain, and Graph Theory for Secure Personalized E-Voting in IoT-Based Distributed Cloud-Fog Social Platforms.
5) Ikhar, A. P. M. D. GRAPH THEORY AND NETWORK ANALYSIS IN SOCIAL NETWORKS.
6) Van Der Hofstad, R. (2024). Random graphs and complex networks (Vol. 2). Cambridge university press.
7) Esmaili, M., & Christensen, K. (2025). Performance modeling of public permissionless blockchains: A survey. ACM Computing Surveys, 57(7), 1-35.
8) Lee, G., Bu, F., Eliassi-Rad, T., & Shin, K. (2025). A survey on hypergraph mining: Patterns, tools, and generators. ACM Computing Surveys, 57(8), 1-36.
9) Ju, W., Yi, S., Wang, Y., Long, Q., Luo, J., Xiao, Z., & Zhang, M. (2024). A survey of data-efficient graph learning. arXiv preprint arXiv:2402.00447.
10) Daneshfar, F., Soleymanbaigi, S., Yamini, P., & Amini, M. S. (2024). A survey on semi-supervised graph clustering. Engineering Applications of Artificial Intelligence, 133, 108215.
11) Musial, K., Wen, J., & Gwyther-Gouriotis, A. (2024). How the use of feature selection methods influences the efficiency and accuracy of complex network simulations. arXiv preprint arXiv:2412.01096.
12) Dai, E., Jin, W., Liu, H., & Wang, S. (2022, February). Towards robust graph neural networks for noisy graphs with sparse labels. In Proceedings of the fifteenth ACM international conference on web search and data mining (pp. 181-191).
13) Lian, Z., Chen, L., Sun, L., Liu, B., & Tao, J. (2023). Gcnet: Graph completion network for incomplete multimodal learning in conversation. IEEE Transactions on pattern analysis and machine intelligence, 45(7), 8419-8432.
14) Lu, Z., Hao, J. K., Benlic, U., & Lesaint, D. (2021). Iterated multilevel simulated annealing for large-scale graph conductance minimization. Information Sciences, 572, 182-199.
15) Xu, Z., Shang, W., Kim, S., Lee, E., & Luo, T. (2025). Quantum annealing-assisted lattice optimization. npj Computational Materials, 11(1), 4.
16) Liao, W., Yang, D., Liu, Q., Jia, Y., Wang, C., & Yang, Z. (2024). Data-driven reactive power optimization of distribution networks via graph attention networks. Journal of Modern Power Systems and Clean Energy, 12(3), 874-885.

17) Zhu, W., Song, G., Wang, L., & Liu, S. (2024). Anchorgt: Efficient and flexible attention architecture for scalable graph transformers. arXiv preprint arXiv:2405.03481.

18) Wang, C., Liu, X., Wang, D., Zhang, H., Chen, B., & Zhou, M. (2024). Scalable Weibull Graph Attention Autoencoder for Modeling Document Networks. arXiv preprint arXiv:2410.09696.

19) El, B., Choudhury, D., Liò, P., & Joshi, C. K. (2025). Towards mechanistic interpretability of graph transformers via attention graphs. arXiv preprint arXiv:2502.12352.

20) Li, H., Liu, J., Zhang, X., Bai, Y., Wang, H., & Mueller, K. (2024). Transforming the latent space of stylegan for real face editing. The Visual Computer, 40(5), 3553-3568.

21) Ju, W., Wang, Y., Qin, Y., Mao, Z., Xiao, Z., Luo, J., ... & Zhang, M. (2024). Towards graph contrastive learning: A survey and beyond. arXiv preprint arXiv:2405.11868.

22) Sun, L., Huang, Z., Wang, Z., Wang, F., Peng, H., & Yu, P. S. (2024, March). Motif-aware riemannian graph neural network with generative-contrastive learning. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 38, No. 8, pp. 9044-9052).

23) Zheng, S., Rao, J., Zhang, J., Zhou, L., Xie, J., Cohen, E., ... & Yang, Y. (2024). Cross-Modal Graph Contrastive Learning with Cellular Images. Advanced Science, 11(32), 2404845.

24) Jung, C., Kwon, G., & Ye, J. C. (2024, March). Patch-wise graph contrastive learning for image translation. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 38, No. 12, pp. 13013-13021).

25) Zhang, X., Tan, Q., Huang, X., & Li, B. (2024). Graph contrastive learning with personalized augmentation. IEEE Transactions on Knowledge and Data Engineering.

26) Xu, R., Wu, G., Wang, W., Gao, X., He, A., & Zhang, Z. (2024). Applying self-supervised learning to network intrusion detection for network flows with graph neural network. Computer Networks, 248, 110495.

27) Zhang, Y., Shou, Y., Meng, T., Ai, W., & Li, K. (2024). A multi-view mask contrastive learning graph convolutional neural network for age estimation. Knowledge and Information Systems, 66(11), 7137-7162.

28) Tan, S., Li, D., Jiang, R., Zhang, Y., & Okumura, M. (2024). Community-invariant graph contrastive learning. arXiv preprint arXiv:2405.01350.

29) Fang, Y., Fan, D., Zha, D., & Tan, Q. (2024, August). Gaugllm: Improving graph contrastive learning for text-attributed graphs with large language models. In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (pp. 747-758).

30) Ma, G. F., Yang, X. H., Long, H., Zhou, Y., & Xu, X. L. (2024). Robust social recommendation based on contrastive learning and dual-stage graph neural network. Neurocomputing, 584, 127597.

31) Tan, J. L. (2024). Performance comparison between generative adversarial networks (GAN) variants in generating comic character images (Doctoral dissertation, UTAR).

32) Barsha, F. L., & Eberle, W. (2025). An in-depth review and analysis of mode collapse in generative adversarial networks. Machine Learning, 114(6), 141.

33) Alsowat, A., Allahyani, R., Alsulami, R., Almaqati, S., Alafif, T., Hawsawi, M., ... & Alsubait, T. Learning to Generate and Predict New Conditional Small Organic Molecules. Available at SSRN 4970369.

34) Ghosh, H., Changyu, C., Sinha, A., & Sural, S. (2024). Heterogeneous Graph Generation: A Hierarchical Approach using Node Feature Pooling. arXiv preprint arXiv:2410.11972.

35) Macedo, B., Ribeiro Vaz, I., & Taveira Gomes, T. (2024). MedGAN: optimized generative adversarial network with graph convolutional networks for novel molecule design. Scientific reports, 14(1), 1212.

36) Davies, A. O., Ajmeri, N. S., & e Silva Filho, T. D. M. (2025). A Comparative Study of GNNs and Rule-Based Methods for Synthetic Social Network Generation. IEEE Access.

37) Gopal, G. N., Kovoor, B. C., & Shailesh, S. (2024). An Improved Deep Learning Network, Addressing Graph Node Imbalance in Social Media Rumor Source Detection. New Generation Computing, 1-35.

38) Sakhamuri, M. R., Henna, S., Creedon, L., & Meehan, K. (2024, June). Molecular Adversarial Generative Graph Network Model for Large-scale Molecular Networks. In 2024 35th Irish Signals and Systems Conference (ISSC) (pp. 01-06). IEEE.

39) Liu, Y., Fan, Y., Li, Z., & Zhang, Q. (2024, November). PreZ-DGGAN: A Drug Graph GAN Based on Pre-Learning of Implicit Variables. In International Conference on Applied Intelligence (pp. 214-225). Singapore: Springer Nature Singapore.

40) Li, C., Tang, H., Zhu, Y., & Yamanishi, Y. (2025). A Reinforcement Learning-Driven Transformer GAN for Molecular Generation. arXiv preprint arXiv:2503.12796.

41) Yang, Y., Xu, K., & Tang, Y. (2024). Gan-based self-supervised message passing graph representation learning. Expert Systems with Applications, 251, 124012.

42) Sun, X., Wang, Z., Lu, Z., & Lu, Z. (2024). Self-supervised graph representations with generative adversarial learning. Neurocomputing, 592, 127786.

43) Wang, P., Liu, Z., Wang, Z., Zhao, Z., Yang, D., & Yan, W. (2024). Graph generative adversarial networks with evolutionary algorithm. Applied Soft Computing, 111981.

44) Hu, Y., Ouyang, S., Yang, Z., Chen, G., Wan, J., Wang, X., & Liu, Y. (2024). Exploring task unification in graph representation learning via generative approach. arXiv preprint arXiv:2403.14340.

45) Li, L., Bi, J., Yang, K., & Luo, F. (2024). Di-GraphGAN: An enhanced adversarial learning framework for accurate spatial-temporal traffic forecasting under data missing scenarios. Information Sciences, 677, 120911.

46) Madushanka, T., & Ichise, R. (2024). Negative sampling in knowledge graph representation learning: A review. arXiv preprint arXiv:2402.19195.

47) Li, L., Yang, P., Lei, H., Wang, B., & Chen, Z. (2024). Multiplex Network Representation Learning Based on Graph Neural Network. Journal of Theory and Practice of Engineering Science, 4(05), 17-24.

48) Yang, H., Kong, Q., & Mao, W. (2024). A deep latent space model for interpretable representation learning on directed graphs. Neurocomputing, 576, 127342.

49) Zhao, J., He, D., Ge, M., Huang, Y., Shan, L., Qin, Y., & Feng, Z. (2024). GA-GGD: Improving semantic discriminability in graph contrastive learning via Generative Adversarial Network. Information Fusion, 110, 102465.

50) Randhawa, R. H., Aslam, N., Alauthman, M., Khalid, M., & Rafiq, H. (2024). Deep reinforcement learning based Evasion Generative Adversarial Network for botnet detection. Future Generation Computer Systems, 150, 294-302.

51) Franceschelli, G., & Musolesi, M. (2024). Reinforcement learning for generative AI: State of the art, opportunities and open research challenges. Journal of Artificial Intelligence Research, 79, 417-446.

52) Hu, Z., Gao, Y., Sun, L., Mae, M., & Imaizumi, T. (2025). A novel reinforcement learning method based on generative adversarial network for air conditioning and energy system control in residential buildings. Energy and Buildings, 336, 115564.

53) Li, M., Su, X., Ma, R., Jiang, T., Li, Z., & Quek, T. Q. (2024). CGGM: A conditional graph generation model with adaptive sparsity for node anomaly detection in IoT networks. arXiv preprint arXiv:2402.17363.

54) Zheng, X., Wu, Y., Pan, Y., Lin, W., Ma, L., & Zhao, J. (2024). DPGAN: A Dual-Path Generative Adversarial Network for Missing Data Imputation in Graphs. arXiv preprint arXiv:2404.17164.

55) Chen, J., Guang, K., Yan, H., Li, Q., & Hao, T. (2024, December). Graph Contrastive Learning with Adversarial Structure Refinement (GCL-ASR). In 2024 IEEE International Conference on Data Mining (ICDM) (pp. 41-50). IEEE.

56) Li, D., Liu, Z., Hu, M., Song, B., & Shan, X. (2024, December). GADIN: Generative Adversarial Denoise Imputation Network for Incomplete Data. In 2024 IEEE International Conference on Data Mining (ICDM) (pp. 221-230). IEEE.

57) Li, H., Li, C., Feng, K., Yuan, Y., Wang, G., & Zha, H. (2024). Robust knowledge adaptation for dynamic graph neural networks. IEEE Transactions on Knowledge and Data Engineering.

58) Wang, X., Zhang, W., Li, G., Zhou, L., Liu, H., & Ye, Z. (2024, September). Deep Graph Convolutional Neural Network based on Adaptive Sampling and Aggregation Strategy. In 2024 2nd International Conference on Machine Vision, Image Processing & Imaging Technology (MVIPIT) (pp. 204-209). IEEE.

59) Li, M., Micheli, A., Wang, Y. G., Pan, S., Lió, P., Gnecco, G. S., & Sanguineti, M. (2024). Guest editorial: Deep neural networks for graphs: Theory, models, algorithms, and applications. IEEE Transactions on Neural Networks and Learning Systems, 35(4), 4367-4372.

60) Ai, W., Shou, Y., Meng, T., & Li, K. (2024). Der-gcn: Dialog and event relation-aware graph convolutional neural network for multimodal dialog emotion recognition. IEEE Transactions on Neural Networks and Learning Systems.