10.48047/jocaaa.2025.34.07.23

Study on Generative Adversarial Networks (GANs) for Drug Discovery

Billa Prahas Reddy
Computer Science and Engineering
Chaitanya Bharathi institute of Technology
Gandipet, Hyderabad, Telangana
Email id:prahasreddybilla@gmail.com

ABSTRACT
Techniques for machine learning (ML) and artificial intelligence (AI) are demonstrating a lot of promise as a foundational tool in the pharmaceutical sector. Deep learning algorithms are being used by researchers to treat a variety of ailments by utilizing the most recent advancements in computing power.In this overview, we focus on three particular areas where the drug discovery and design process has lately benefited greatly from deep learning techniques, such as GAN frameworks. The first part of the study focuses on drug discovery and design studies that have evaluated a single application using GAN techniques, such as molecular de novo design. This paper introduces a method for training GANs that, drawing on ideas from Genetic Algorithms, encourages incremental exploration while mitigating the effects of mode collapse. Instead of using training data samples, our method uses genuine samples from the generator. We take into account recombination during replacement, guided selection, and random selection. We show that training data updates significantly outperform the standard method by tracking the quantity of novel compounds generated throughout training, which expands the potential usage of GANs in drug discovery. By modifying hyperparameters and assessing drug-likeness qualities like pharmacokinetics, toxicity, and synthetic accessibility, a deep learning model known as MedGAN—which was created to produce novel quinoline-scaffold compounds from intricate chemical graphs—is refined and optimized. While maintaining chirality and atomic charge, the ideal model produced 92% quinolines, 93% novel compounds, 95% unique, 62% totally linked compounds, and 25% authentic compounds.

Keywords: machine learning (ML) and artificial intelligence (AI), GANs, drug discovery.

I. INTRODUCTION

The multidisciplinary domains of AI, ML, and medication discovery and development have recently witnessed remarkable advancements in research [1]. Algorithms developed using data and AI/ML can be useful across the drug development pipeline, including target prediction, screening, discovery, preclinical trials, and clinical trials [2]. The pharmaceutical industry is a prime example of this. Innovations in deep learning algorithms and other forms of machine learning have recently demonstrated encouraging outcomes in the field of medication research and innovation. During the preclinical stage of drug development, researchers can utilize single-cell RNA sequencing (scRNA-seq) technologies to decrease the dimensionality of single-cell data in order to identify cell-specific biomarkers. Some deep learning approaches used for this purpose include the deep variational autoencoder [3]. One further intriguing use of deep learning is in medication development and screening processes, where deep variational autoencoders are used to create new chemical compounds. Deep learning techniques are expected to be crucial in the discovery of novel medications soon due to their wide range of applications [4]. Utilizing cutting-edge AI and ML models, deep learning techniques essentially build data representations utilizing multiple levels of abstraction in a hierarchical fashion. To build the hierarchical representation, for instance, one can use artificial neural networks [5]. In order to make things easier to understand, deep learning approaches solve optimal prediction problems by utilizing computer programs that employ multiple layers of artificial neural networks rather than just one. Modern computing power, including GPUs for general-purpose computing, has numerous applications in the pharmaceutical industry, including deep learning [6]. In order to tackle the complex problems we face in drug design and discovery today, software tools within deep learning frameworks are greatly needed to perform various activities related to drug research. Molecular de novo design, using deep learning frameworks to reduce the dimensionality of single-cell data in preclinical development, predicting the activity and properties of compounds, analyzing reactions, predicting synthesis, and analyzing biological images are all applications in drug discovery and design [7]. A new method called the generative adversarial network (GAN) architecture has recently gained a lot of attention in the AI and

10.48047/jocaaa.2025.34.07.23

ML sectors due to advancements in deep learning frameworks [8]. The GAN architecture has the potential to revolutionize numerous fields, including image processing, language processing, medication discovery and development, and countless more. Research into drug design and development has also profited from using the GAN pattern. One of the many important ongoing endeavors in the field of drug design and discovery is molecular de novo design, which takes the GAN architecture into account [9]. To help find and create new chemicals for cancer treatment, one GAN-based framework, deep adversarial autoencoder structure, has combined chemical and biological data. Specifically, the dimensionality of data from single-cell RNA sequencing has been effectively reduced using the deep adversarial variational autoencoder structure during the preclinical stage of drug discovery [10]. Drug design and discovery leverage two GAN-based frameworks: the deep adversarial autoencoder and the deep adversarial variational autoencoder. The details of these structures are now revealed. Recent years have seen deep learning's rise to prominence as a potentially game-changing method for increasing the capacity of massive virtual screening libraries, revealing previously unseen patterns and interconnections, and ultimately, finding novel, potentially bioactive compounds within massive molecule databases via docking, screening, or de novo design. In order to make drug candidate research more efficient and accurate, generative AI can rapidly access large chemical libraries, learn from previous mistakes, and predict new binding poses or compound combinations using these patterns [11]. A number of applications have been explored in the past, including antibiotic discovery through existing molecule drug repurposing and drug optimization and design through the use of generative adversarial networks, autoencoders, reinforcement learning, and recursive neural networks to produce new compounds with minimal waste of resources [12]. An improved generative architecture based on an enhanced Generative Adversarial Network (MedGAN) is the end result of this research. Its development and publication are the primary objectives of this work. Here, we train two models in parallel: one uses multilayer perceptrons to map noise, and it uses backpropagation and dropout methods; the other uses a training data sample to determine the likelihood. To represent molecules as complex graphs with various relations, such as atomic type, chirality, charge, and connections (edges) between nodes (atoms) and their attributes (features), graph convolutional networks (GCNs) are employed. According to [13], a GCN and a Wasserstein GAN (WGAN) were used to optimize the GAN model. Problems like mode collapse are eliminated by the method's steady training dynamics, which use the Wasserstein distance as a loss function. The critic's output doesn't alter suddenly in reaction to changes in input thanks to a gradient penalty, and it employs GCN layers to improve representations of edges and nodes. Small molecular graphs have been produced from previous research that used WGAN with GCN to create molecules. The complexity of the drug-like compounds limits their size, complexity, and performance. Choosing a single scaffold that is known to have biological interest is one way a specialized method can further improve drug discovery efficiency. A more effective and economical generative model is the outcome of this approach, which decreases the latent space needed for learning and addresses a common pattern [14]. This expands upon the multifaceted terrain of medication discovery and the growing significance of generative models. The unique chemical features and wide range of biological activities of quinoline scaffold molecules make them excellent candidates. Their pyridine-like nitrogen and polycyclic aromatic rings are essential building blocks of many biologically active compounds. These rings are capable of undergoing a wide range of electrophilic substitutions, resulting in an abundance of varied molecules with unique stereochemistry, many of which have demonstrated therapeutic efficacy. It is well-known that quinoline and its derivatives have antibacterial, antiviral, anti-inflammatory, and anticancer effects, making them a potential candidate for the development of novel medicines. This structure is intriguing and might be useful for future drug design studies because of its several action mechanisms, such as restricting angiogenesis, controlling cell migration, and reducing cell proliferation through cell cycle arrest and death [15]. The findings point to a complex network of interdependencies among several generative AI components, including diffusion models, GPT, BERT, transformers, and LLMs. Their common denominator is the transformer architecture, a special kind of neural network that does exceptionally well with NLP. Table 1 compares a number of GAI technologies:

TABLE 1. Various GAI Frameworks.

10.48047/jocaaa.2025.34.07.23

Model	Description	Example Applications
Transformer	A type of neural network developed by Google	Machine translation, text generation, question answering
LLM	Type of neural network that is trained on a massive dataset of text	Generation and translation of text, question answering, analysis of sentiment, named entity recognition
GPT	A LLM developed by OpenAI	Capable of generating and translating languages, writing various unique content, and providing insightful answers to users' prompts
BERT	A LLM developed by Google AI	Particularly good at understanding the situation of words in a sentence. It can be used for sentiment analysis, named entity recognition, and question answering
Diffusion model	Type of LLM that is trained on a dataset of images	It can generate new images similar to those in the dataset.
Flow-based model	Type of generative model that explicitly models a probability distribution by leveraging normalizing flow	Image generation, density estimation, anomaly detection

2. LITERATURE REVIEW

2.1 Generative Adversarial Network (GAN) Architecture

A critical view of GAN architecture as a generative model type was first put out in [16]. The GAN architecture has subsequently gained immense popularity within the AI and ML fields. The GAN architecture is now on the cutting edge of computer vision and image processing, thanks to all the recent breakthroughs, including picture generation. The deluge of publications covering many GAN architecture variants in various scientific and technical domains makes keeping up with the latest trend extremely difficult [17]. The benefits of the GAN architecture are as follows. To begin, empirical investigations show that GAN architecture outperforms competing generative methods in many cases. The second benefit is that the sample creation is significantly accelerated by the GAN design's ability to execute the sampling procedure in parallel. Thirdly, the GAN architecture can be executed without the need for real-world data distributions or mathematical requirements [18].

2.2. Brief Description of the GAN Architecture

The two primary parts of a GAN architecture are the generating network and the discriminative network modules, as shown in Figure 1. The generating and discriminative network modules are built on top of two concurrently trained multi-layer artificial neural networks. The discriminative network module accepts both true and false instances and determines if the input is real or not, as opposed to the generative network module that is trained to create fake instances using the latent variable. The discriminative network module will generate a more likely prediction whenever it detects an occurrence with a higher likelihood of being real. Training the discriminative and generative network modules simultaneously increases the error probability. Stated differently, the discriminative and producing networks work together to achieve their objectives. The generating and discriminative network modules can thus play an adversarial game thanks to the GAN design.

10.48047/jocaaa.2025.34.07.23

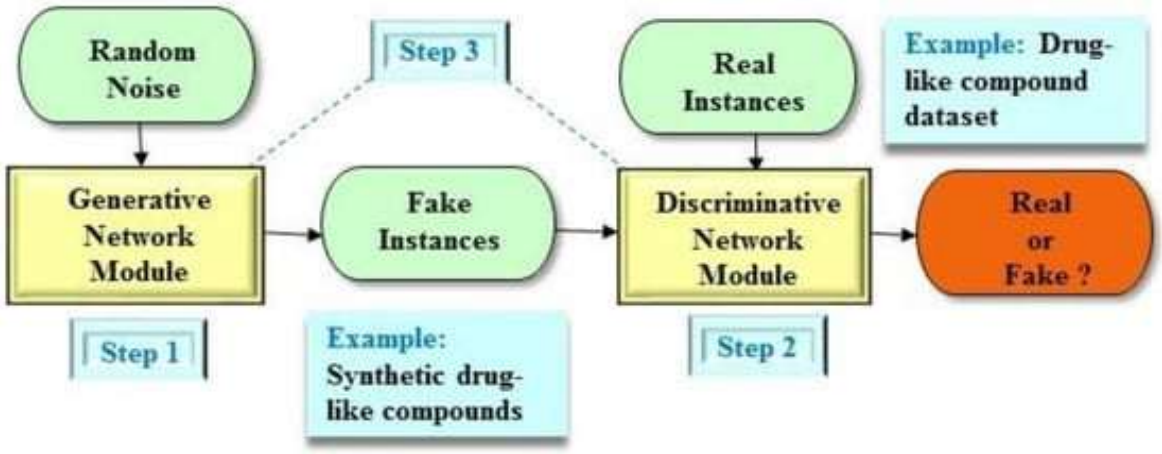

Figure 1. A generative adversarial network (GAN) architectural example.

2.3. Applications of the GAN Architecture

Applications for the GAN architecture can be found in many different fields, such as video processing, speech, music, medical imaging, and computer vision. Also, the GAN architecture has found applications in numerous domains, including as chemoinformatics and molecular informatics in chemistry and bioinformatics, drug discovery, medical informatics, multi-omics, and bioinformatics in biology [19].

2.4. Variants of the GAN Architecture

Various implementations of the GAN architecture result in a multiplicity of GAN-based frameworks. Here, we present three variations: the structures of conditional GANs, deep adversarial autoencoders, and Wasserstein GANs. The most recent evaluations provided by [20] might be consulted by readers who are interested in knowing more about further GAN architectural variations.

2.4.1. Wasserstein GAN

As a solution to the GAN training instability, the Wasserstein GAN structure was proposed in [21]. The Earth-Mover distance, also called the Wasserstein distance, is a new way of measuring distance that this building employs. Theoretically, determining the distance between non-overlapping distributions is not possible using the Jensen-Shannon divergence that was originally employed in GAN architecture.

2.4.2. Conditional GAN

In the conditional GAN architecture proposed in [22], class labels are one piece of auxiliary data that is dependent on the discriminative and generative network modules. Supervised conditional GANs use a conditional variable, like class labels, to direct learning, as opposed to the unsupervised GAN design. The generative and discriminative network modules both require the conditional variable as an additional input. A discriminative network module learns the relationship between examples and class labels, whereas a generative network module learns to create hypothetical instances using the latent variable and class labels. One significant advantage of GAI-driven medicine discovery is the capacity of GAI models to generate whole novel molecules. De novo molecular generation is the process of producing new molecule structures with certain characteristics. Consequently, attribute-predicting neural networks are widely used in GAI models. A system called ReLeaSE was introduced in a sample research [23] to demonstrate the idea. The generative and discriminative network modules both require the conditional variable as an additional input. A discriminative network module learns the relationship between examples and class labels, whereas a generative network module learns to create hypothetical instances using the latent variable and class labels. One significant advantage of GAI-driven medicine discovery is the capacity of GAI models to generate whole novel molecules. Nevertheless, Popova et al.'s GAI theory was not supported by any experimental data. Our experimental results show that GAI model training on natural products yielded the compounds that act as novel retinoid X receptor (RXR) modulators [24]. To find new DDR1 kinase inhibitors that can fight fibrosis, Insilico Medicine of Hong Kong/New York created a GAI model called GENTRL (Generative Tensorial Reinforcement Learning). To ensure the efficacy of the GENTRL chemicals, they subjected them to biological testing. Being the first of its type to progress from insilico to preclinical in just 21 days, their work stood out. By integrating RL techniques such as experience replay, real-time incentives, policy gradient algorithm, and transfer learning, Korshunova et al. [25] were able to develop novel, experimentally-validated EGFR inhibitors. Two networks, the encoder and the decoder, are trained to produce autoencoders. A low-dimensional latent vector is

10.48047/jocaaa.2025.34.07.23

created from the input data by the encoder, and then the input data is returned to it by the decoder. To create latent spaces, autoencoders duplicate the input. Molecular structure (in latent space) cannot be continuously described by such simple autoencoders. VAEs were developed in response to the overfitting and discontinuity in the original autoencoders. One of the most popular GAI approaches is unsupervised data compression using VAEs since it is effective with different types of input data and can handle a lot of different scenarios. Complex data kinds, such as faces, handwritten numbers, and segmentation, can be successfully generated with VAEs [26]. VAEs are widely employed in the hunt for new compounds having medicinal properties. In actuality, VAEs are the primary target of current GAI-based medication research initiatives. An encoding network and a decoding network are the two main parts of a VAE. Encoder networks reduce the dimensionality of data by compressing it into latent space after first transforming it into a probability distribution. Using a latent space sample, the decoder network recreates the input data. Finding the optimal parameters is the goal of the VAE, which is to increase the likelihood of correctly recreating the input data. By enhancing their ability to generate realistic samples in the latent space, VAEs can infer new data. To optimize the similarity between encoded molecules and a previous distribution in latent space, VAEs strive to minimize reconstruction loss during training, which is important in drug development. Through sampling from the learnt latent space, VAEs have the potential to create new compounds that share structural and chemical properties with the training data.

3. METHODS

New quinoline-like compounds were generated using deep learning methodology and data management techniques, which are detailed in this section. Improving and optimizing the Wasserstein Generative Adversarial Networks (WGANs) design is the main objective of this research.

Building a novel GAN model that could produce complex scaffold-specific chemicals that were legitimate, distinct, and free of attention or reinforcement processes was the goal. By learning fundamental patterns, such the molecular scaffold in the quinoline structure, this strategy improves the model's performance and sets the stage for future generalization of other scaffold-specific structures or focused biological functions.

3.1 MedGAN generative model

Two issues that might make training GANs difficult are mode collapse and unpredictable training dynamics. When the GAN produces homogeneous samples, a mode collapse happens, and when the Generator and Discriminator fluctuate without enhancing the model, unstable training dynamics emerge. Fig. 2 shows how a Graph Convolutional Network and tiny molecular graphs generated by Wasserstein GANs address these issues. By combining a gradient penalty function with an alternative loss function called the Wasserstein distance, or Earth Moving Distance, WGAN improves the stability and reliability of the training process. This creates a more gradual gradient from which the generator learns, decreasing the probability of training becoming stuck. The goal of this min-max game is to minimize the Wasserstein distance, and the discriminator and generator are trained simultaneously using the WGAN framework. Thus, the generator gets better at making realistic graphs over time, while the discriminator gets better at telling actual graphs apart from fakes. In order to process random noise and produce initial node and edge representations, the generator network first employs thick layers. Then, by considering the features and connections inside the graph, many Graph Convolutional Network (GCN) layers are used to enhance them.

10.48047/jocaaa.2025.34.07.23

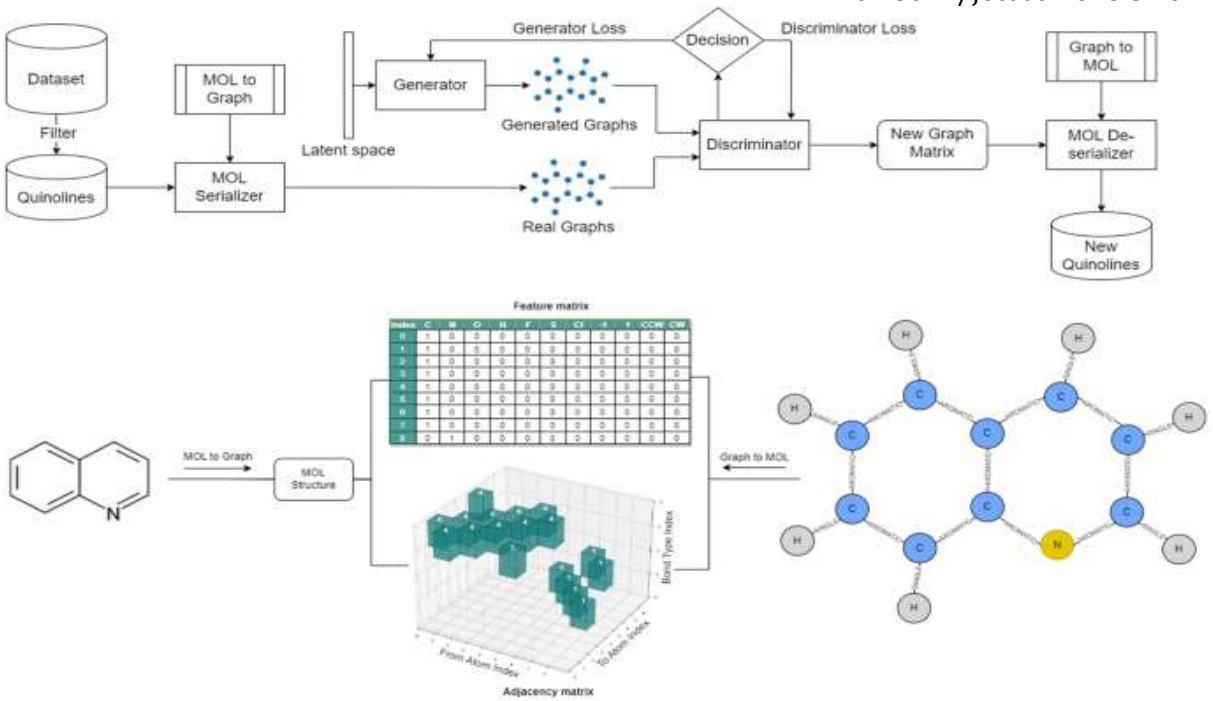

Figure 2. The Architecture of the MedGAN Generative Model. This is an example of the MedGAN model in action, which uses Graph Convolutional Networks and Wasserstein GANs to generate molecular graphs from a sample of molecules. The process begins with ZINC15 or PubChem data and continues via serialization, learning, deserialization, and SMILES to Mol conversion. Through optimization using specified configurations, quinoline-scaffold molecules are generated. upplementary Data 3 contains further information.Synthetically produced graphs with nonhomogeneous relations have their final node and edge attributes computed after multiple iterations of processing these refined representations through an additional set of dense layers. In order to process the input graph, the discriminator network uses a GCN. It begins by collecting features from nodes and edges, which are then passed via GCN layers to create a representation of the graph at the graph level that takes into account both the local features and the general structure of the network. A scalar value, indicating the input graph's legitimacy, was generated by passing this representation through a dense layer.

3.2 Models

Using pytorch, we built a GAN with four completely connected layers for the discriminator and the generator. The 8-dimensional randomly distributed vectors were fed into the generator. The generator generated an adjacency matrix with elements that showed the atom type on the on-diagonal side and the bond order on the off-diagonal side. The discriminator generated a single real number after receiving the single hot representation from the adjacency matrix. We employed the standard GAN minimax loss for training. For every run, a learning rate of 10−4 was utilized with the Adam optimizer.

3.3 Updates to training data

The training intervals for all models were 5 epochs. Ten thousand samples were taken from the generator for each period. We combined samples that were new and valid spanning all five epochs. Next, the initial training data was subjected to a replacement method, which may be either random or drug-based. To implement random replacement, a random selection was made from the current training samples and then substituted. The current samples were arranged according to their drug-likeness score in order to facilitate drug replacement. The only time an update was issued was when a fresh sample outscored the old one. The method for updating drug-likeness was also applied to other measures, such as synthesizability and solubility. We also took recombination into consideration for both replacement options. Half of the 10,000 generator sample was crossed over with the pre-existing training data during recombination. Crossover involves inserting a tiny amount of the existing training data into an entirely new adjacency matrix. A uniform random sample of an integer between 1 and the adjacency matrix's length is then selected. To make a new possible molecule, just swap out the corresponding section of the copied matrix (say, the first five rows and columns) with the corresponding section of the newly formed matrix.

10.48047/jocaaa.2025.34.07.23

For the drug replacement strategy, a weighted sample of the training set was chosen according to the drug score. The softmax of the scores for the current measures was used to calculate the weights of the training data. As mentioned before, the replacement process then continued. The method for updating drug-likeness was also applied to other measures, such as synthesizability and solubility.

4. RESULTS AND STUDY

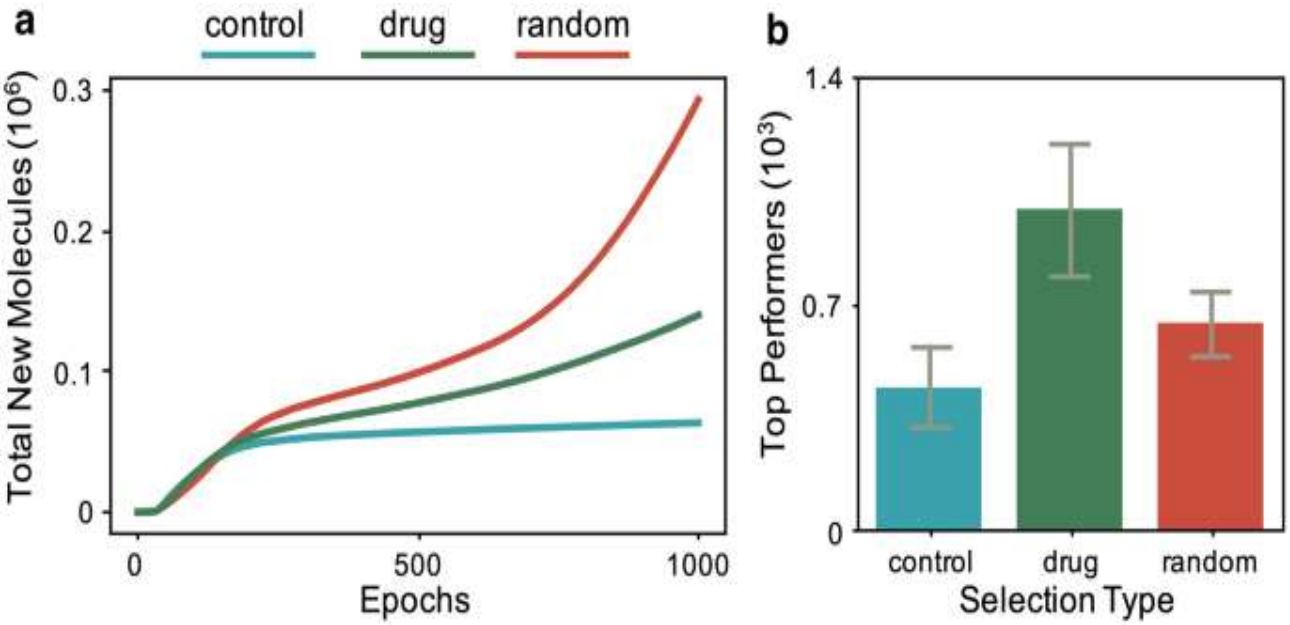

Fig 3: Generated novel compounds to support various replacement plans. The blue control set has fixed training data. If you choose random (red), the generator's molecules will substitute for those in the training data at random. Only molecular replacements from the generator with a higher drug-likeness score will be used for drug (green). a While random and medication replacement techniques keep producing molecules throughout training, control stops producing a significant quantity of new molecules at a certain point. Each selection type's average over three training runs is displayed in the plot. b Compared to random, medication makes fewer new molecules overall, but it produces more high-performing compounds. Each selection type's plot displays an average across three runs, with error bars displaying one standard deviation.

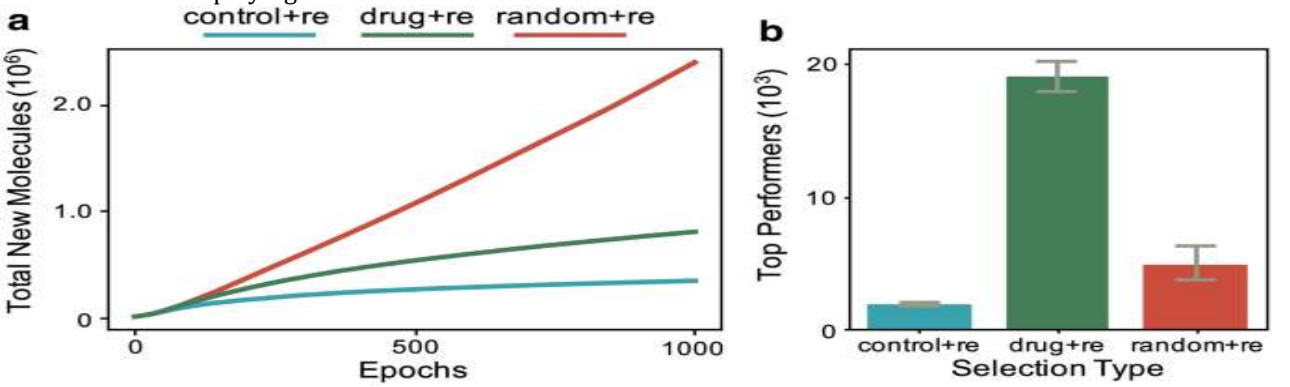

Fig 4: Utilizing recombination, novel molecules are generated for use in various replacement schemes. The training data remains unchanged for control+re (blue). The red value, random+re, indicates that the generator's molecules will substitute for training data molecules at random. In drug+re (green), only generator molecules with a higher drug-likeness score are used to replace training samples. a Random and medication replacement techniques perform better than control as training goes on, just like in the case without recombination. Each selection type's average over three training runs is displayed in the plot. b While drug+re does not develop as many novel compounds as random+re, it does produce more high-performing molecules. Each selection type's plot displays

10.48047/jocaaa.2025.34.07.23

an average across three runs, with error bars displaying one standard deviation.Figures 3 and 4. Take note of how the metric-specific selection technique produces the highest number of top achievers across all metrics.

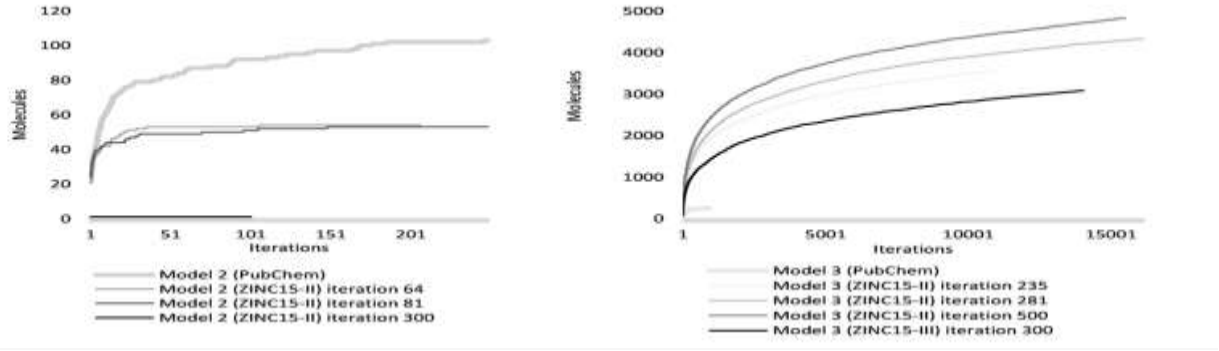

Figure 5. Model generation performance.

The process of creating new, authentic, and distinct quinoline molecules came to a halt after 100 iterations. Model 3 created 253 molecules during optimization, but Model 2 only managed to make 115 was shown in figure 5. whereas using ZINC15-II for training, Model 3 produced 4831 unique quinoline molecules, whereas using ZINC15-III resulted in 3020 unique quinoline molecules during the fine-tuning step. Model 2 was unable to produce fully coupled molecules after producing 54 molecules. There was a noticeable disparity in model performance (a 22% increase in molecule generation at iteration 281), a somewhat different outcome for connected validity (0.64 and 0.68, respectively), and similar validity (0.19 at iterations 235 and 281). These findings pointed to a non-linear relationship between training performance metrics and molecule diversity.

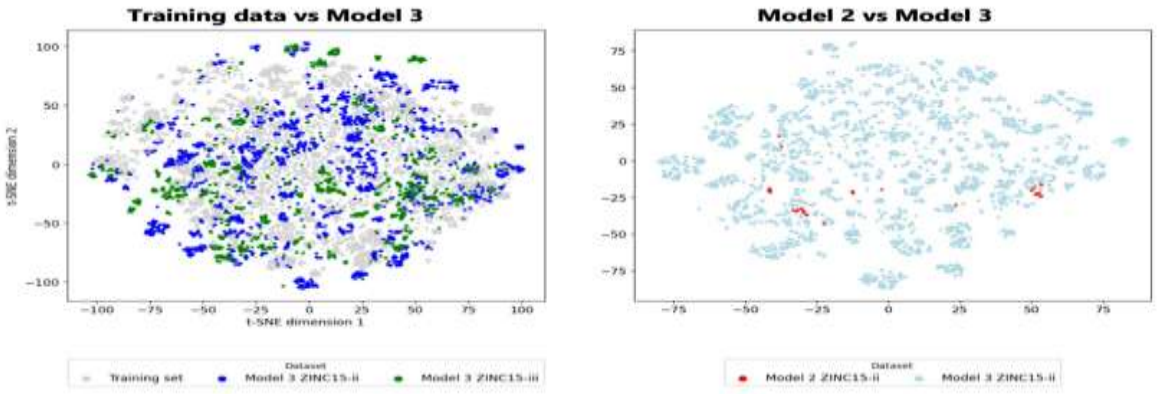

Figure 6. molecules to be seen using t-SNE. Both Model 2 (subset ZINC15-ii, red) and Model 3 (subsets ZINC15-ii, blue, and ZINC15-iii, green) were responsible for producing molecules, in contrast to the 10,000 randomly picked samples from the training data (grey).Quinoline molecules could not be produced by either Model 1 or the basic model. Models 2 and 3 could generate structurally similar and fully connected molecules depending on a number of factors, including the type and amount of training data, activation function, optimizer, learning rate, latent space, and neuron units of the Generator and Discriminator. With fewer units of latent space and neurons in the generator and discriminator, as well as the RMSProp optimizer, the second model put validity and speed first. In contrast to Model 3, Models 2 and 4 illustrate the pros and cons of using the RMSProp optimizer, a larger latent space, and more neuron units to produce diverse and fully connected molecules. Furthermore, by evaluating the generative capability at various iterations in Model 3, we were able to gain a clearer picture of how performance measurements relate to model output, which in turn shed light on how effective generation is shown in figure 6.

CONCLUSION

One effective method for exploring chemical space for target functionality is to use generative machine learning models, such as GANs. In this paper, we laid up a plan to use incremental data updates to encourage search outside of the initial training set. Our approach takes a step towards automating the frequently manual mutation rules; it is based on the well acknowledged ideas of genetic algorithm selection and recombination. Our findings imply that, in comparison to a static training set, data updates allow for the exploration of a greater number of compounds,

10.48047/jocaaa.2025.34.07.23

resulting in an increase in high-performing candidates. Finally, our study addressed the unique requirements of drug discovery while integrating the intricacies of generative modeling. The insights gained, together with the immediate effects, provide a deeper understanding of model behaviors, optimization tactics, and the complexities of molecular design. An improved and more personalized approach to medication development is an immediate outcome of this work's contributions to the current body of knowledge.

REFERENCES

1. DiMasi JA, Grabowski HG, Hansen RW (2016) Innovation in the pharmaceutical industry: new estimates of R&D costs. J Health Econ 47:20–33
2. Rashid MBMA (2021) Artificial intelligence effecting a paradigm shift in drug development. SLAS Technol 26(1):3–15
3. Xue H, Li J, Xie H, Wang Y (2018) Review of drug repositioning approaches and resources. Int J Biol Sci 14(10):1232
4. Polishchuk PG, Madzhidov TI, Varnek A (2013) Estimation of the size of drug-like chemical space based on gdb-17 data. J Comput Aided Mol Des 27(8):675–679
5. Rodrigues T, Reker D, Schneider P, Schneider G (2016) Counting on natural products for drug design. Nat Chem 8(6):531
6. Mouchlis VD, Afantitis A, Serra A, Fratello M, Papadiamantis AG, Aidinis V, Lynch I, Greco D, Melagraki G (2021) Advances in de novo drug design: from conventional to machine learning methods. Int J Mol Sci 22(4):1676
7. Schneider G, Fechner U (2005) Computer-based de novo design of drug-like molecules. Nat Rev Drug Discov 4(8):649–663
8. Gómez-Bombarelli R, Wei JN, Duvenaud D, Hernández-Lobato JM, Sánchez-Lengeling B, Sheberla D, Aguilera-Iparraguirre J, Hirzel TD, Adams RP, Aspuru-Guzik A (2018) Automatic chemical design using a data-driven continuous representation of molecules. ACS Cent Sci 4(2):268–276
9. Gupta A, Müller AT, Huisman BJH, Fuchs JA, Schneider P, Schneider G (2018) Generative recurrent networks for de novo drug design. Mol Inform 37(1–2):1700111
10. Segler MHS, Kogej T, Tyrchan C, Waller MP (2018) Generating focused molecule libraries for drug discovery with recurrent neural networks. ACS Cent Sci 4(1):120–131
11. Williams RJ (1992) Simple statistical gradient-following algorithms for connectionist reinforcement learning. Mach Learn 8(3–4):229–256
12. Olivecrona M, Blaschke T, Engkvist O, Chen H (2017) Molecular de novo design through deep reinforcement learning. J Cheminform. https://doi.org/10.1186/s13321-017-0235-x
13. Liu X, Ye K, van Vlijmen HWT, IJzerman AP, Van Westen GJP (2019) An exploration strategy improves the diversity of de novo ligands using deep reinforcement learning: a case for the adenosine A2a receptor. J Cheminform 11(1):1–16
14. Popova M, Isayev O, Tropsha A (2018) Deep reinforcement learning for de novo drug design. Sci Adv 4(7):7885
15. Zheng S, Yan X, Gu Q, Yang Y, Du Y, Lu Y, Xu J (2019) Qbmg: quasi-biogenic molecule generator with deep recurrent neural network. J Cheminform 11(1):1–12
16. Guimaraes GL, Sanchez-Lengeling B, Outeiral C, Farias PLC, Aspuru-Guzik A (2017) Objective-reinforced generative adversarial networks (ORGAN) for sequence generation models. arXiv preprint arXiv:1705.10843
17. Yu L, Zhang W, Wang J, Yu Y (2017) SeqGAN: Sequence generative adversarial nets with policy gradient. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31
18. Blaschke T, Olivecrona M, Engkvist O, Bajorath J, Chen H (2018) Application of generative autoencoder in de novo molecular design. Mol Inform 37(1–2):1700123
19. Gui J, Sun Z, Wen Y, Tao D, Ye J (2021) A review on generative adversarial networks: algorithms, theory, and applications. In: IEEE Transactions on Knowledge and Data Engineering
20. Putin E, Asadulaev A, Ivanenkov Y, Aladinskiy V, Sanchez-Lengeling B, Aspuru-Guzik A, Zhavoronkov A (2018) Reinforced adversarial neural computer for de novo molecular design. J Chem Inform Model 58(6):1194–1204

10.48047/jocaaa.2025.34.07.23

21. Prykhodko O, Johansson SV, Kotsias P, Arús-Pous J, Bjerrum EJ, Engkvist O, Chen H (2019) A de novo molecular generation method using latent vector based generative adversarial network. J Cheminform 11(1):1–13
22. Xiong J, Xiong Z, Chen K, Jiang H, Zheng M (2021) Graph neural networks for automated de novo drug design. Drug Discov Today 26(6):1382–1393
23. Grechishnikova D (2021) Transformer neural network for protein-specific de novo drug generation as a machine translation problem. Sci Rep 11(1):1–13
24. Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using RNN encoder–decoder for statistical machine translation. arXiv preprint arXiv:1406.1078
25. Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. arXiv preprint arXiv:1409.3215
26. Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V, Courville AC (2017) Improved training of Wasserstein GANs. Advances in neural information processing systems. 30