AI in Drug Discovery and Development

Prof. Rajnish Kumar
Dept. of Pharmaceutical Engineering and Technology
IIT-(BHU), Varanasi

Week-09
Lecture-42

Welcome to the course "AI in Drug Discovery and Development." In this session, we will talk about deep generative models for drug design. So, in the earlier session, we saw an introduction to deep generative modeling and why de novo drug design and generative AI are important in drug discovery and development. And in this case, we will talk about some of the models that are being extensively used for de novo drug design using deep generative modeling. So, by the end of this lecture, you will be able to understand the role of deep generative models in accelerating drug discovery and design. Identify and differentiate key generative architectures such as VAEs, variational autoencoders, GANs, RNNs, and transformer models used in molecular generation.

Evaluate the benefits and limitations of applying deep generative models to real-world drug development. So, what are those deep generative models? We saw them in the earlier session, too. However, these deep generative models are designed to learn complex data distributions and generate new realistic samples, and they are useful in fields like image and text generation that we have observed. With the emergence of ChatGPT, Gemini, and all those large language models or transformer-based models that can generate text, which can even generate images, music, sound, and all sorts of creative things they can do.

And they are crucial in drug discovery as well because they can be trained to develop or generate molecules. So, we have seen the importance of de novo drug design as well, because traditional drug discovery is really time-consuming and cost-intensive as well. And then, de novo drug design using deep generative models enables the exploration of vast chemical space efficiently. Where we saw that virtual screening, which is commonly used to identify hit molecules. So, you know, it is not exploring a huge amount of the chemical space because it is limited to what we actually have: those molecules.

So, it is limited to the molecules that we already have in stock or that can be synthesized, but a generative model can explore the space beyond that as well. So, these models can design novel drug-like molecules with desired properties, accelerating the lead discovery process as well. So, during lead development or lead optimization, when we wanted to convert one molecule into a potent, selective, and non-toxic molecule. So, in that case, these deep generative models are highly useful because we can start with the initial molecule that we have in our hands and try to optimize it while retaining all those

properties; it can also be called multi-parameter optimization. They are increasingly useful for de novo drug design in generating novel molecular structures with desired biological activities and properties.

Okay, so now we mainly have two kinds of generative models. One is the chemical language model, which is the sequence-based model that generates molecules as strings; then we have the graph-based models. The models that represent molecules as graphs capture atoms as nodes and bonds as edges. Now we will have a look at both these chemical language models and graph-based model. So, looking at the chemical language models, you can see here, for example, that this is the pyridine structure.

So, now this pyridine can actually be represented as a smile. This is the SMILES string for the pyridine. So, now the generative process will start by adding the characters one by one, and then it can finally build the smile string, which can be converted into a molecule. So, these chemical language models, if we talk about the representation, represent these molecules as sequences in chemical language models; for example, they can be a SMILES string. The architecture of these models is based on sequence learning, as they mainly use recurrent neural networks, LSTMs, or transformers, which are all based on the sequence learning paradigm.

And then the training data requires large datasets of molecular sequences and a large number of molecular structures in the form of SMILES. We need to train those models like ChatGPT is trained on the transformer model, which is trained on text. So, it can actually generate beautiful text. Likewise, we can generate these deep neural networks on the SMILES structure. So that they can learn the grammar and syntax of SMILES, they can actually generate those new SMILES.

So, the model workflow looks like this: if we talk about the SMILE-based generation. The model predicts the next character in a SMILES sequence and generates new molecules one token at a time, and some of the strengths of these chemical language models are that they are sequence-based. So, they are easier to train on the existing data, and pre-trained models like GPT-2 can be adapted for drug discovery, and the SMILES encoding itself is, you know, an advantage because it can capture many chemical features through sequence-to-sequence mapping as well. However, there are some limitations, such as the discrete representation, because the SMILES strings limit the 3D structure, and the stereochemical representation, because SMILES are just a 1D string, actually. So, it does not account for the 3D structure, such as how a molecule interacts with the binding pocket of a receptor and why it is acting, and which of those pharmacophoric features are present that are responsible for the activity.

So, we wanted to add that information, but it is not possible in these chemical language models, actually. And then there is a loss of 3D context because it lacks an inherent 3D molecular structure. Some examples of these chemical language models are MOL-GPT; it is a generative model for generating drug-like molecules. And then ChemBerta is also an example of a chemical language model, which is a bird-based model for molecular property prediction and generation. And then there are some applications.

They can be used for, you know, the generation of drug-like molecules, for chemical reaction prediction as well, and for virtual screening, exploring chemical space via the SMILES-based representation. On the other hand, we have those graph-based models, where molecules are represented as graphs: atoms are represented as nodes, and bonds are represented as edges. So, these models are based on, you know, graph neural networks like GCN, GraphSAGE, or message-passing networks. So, they require talking about the training data. So, they require graph-based data sets with features like nodes, which are the actual properties, and edge types.

So, one of the limitations, or we can say one of the challenges with these graph-based models, is the data itself, because we have to convert all those molecules into graphs, and then we are converting them into graphs. So, they consume a lot of storage as well because those files become really big. So, training and running these graph-based neural nets for generative modeling require a lot of computational power as well as storage. Talking about the model workflow, we use graph-based generation where graphs are iteratively generated or modified by learning molecular features such as atom connectivity, bond types, and the 3D structure. One of the advantages is that it can also take into account the 3D structure of the molecule.

And then the strengths are that you can have a precise representation of molecular structure because it can capture atomic and bond-level features, and it has 3D structure compatibility as well. We can integrate 3D molecular properties to enhance molecule generation for drug discovery, and of course, it has enhanced learning, allowing it to incorporate various molecular features, such as electronic properties, as well. However, there are some limitations as well, such as the complexity, because graphs are harder to manipulate than sequences. So, they require sophisticated learning approaches, and they are computationally intensive, as I said; they need more resources for training and inference, especially a lot of computational storage. So, some of the examples are DeepChem, which is a toolkit for applying graph-based models to molecular property prediction and generation, and then we have the graph MVP.

And then there is a graph invented as well, which comes from AstraZeneca. So, it is a graph generative model for generating molecular structures. So, what we can do with these

graph-based models is again to generate new molecules using the de novo approach. However, we can predict molecular properties using these graph-based neural networks, including bioactivity, toxicity, solubility, etc. And we can do de novo drug design, generating novel molecules directly from the graph representation.

And we can do the molecule optimization as well, where we can optimize the compounds for the desired properties. Okay, so if you look at the evolution of graph-based generative models, it started, you know, in 2017 when the first graph-based model was developed, and then we had the graph VAE, the graph net, and then JT-VAE. And then we had the first GAN-based model around 2018 and 2019. So, the time between 2018 and 2019 saw huge development in these graph-based generative models. And then recently we saw, you know, all these graphs AF and MoFlow, and delinker that are being used for, you know, the linker design and then scaffold-based designs.

So, all those developments happened between 2019 and 2020. So, this is a little bit like the old. However, if you look now at the period from 2020 to 2025, So, we have again got a lot of new development where we can now include the structural features, the 3D structural features, and drug target interaction data into these models as well, where we can do the structure-based de novo generative modeling as well. So, some of the models that are being used extensively in de novo drug design are generative adversarial networks, GANs, transformer-based models, recurrent neural nets, variational autoencoders, and graph neural networks. So, some of those we have already covered in the earlier sessions, like recurrent neural networks or graph neural networks, but we will take a quick look at them, including what they can do, how they are, and how they actually work.

So, let's come to the recurrent neural networks for de novo drug design. So, these RNNs are neural networks designed for processing sequential data. So, in drug design, RNNs generate molecular structures step by step, learning patterns from sequences like SMILES and other molecular representations. So, if we talk about the architecture of these RNNs, they have, you know, the sequential generation where they generate the molecules one token at a time, and that is the atom or the bond based on the previous tokens. And then they have a hidden state, it maintains an internal state updated at each step to capture the earlier information.

And then we can also use those LSTMs, which are a variant that helps address the vanishing gradient issue and improve long-range dependencies. So, they can be either smile-based RNNs, which are, you know, the chemical language-based models, or graph-based RNNs. So, the smiles-based RNN generates molecules as character sequences using RNN architectures like LSTMs or GRUs, and then we have the graph-based RNNs. So, some RNNs generate molecular graphs by predicting atoms and bonds sequentially. So,

the advantages of RNN are the sequential control, which allows for control over the generation process, ensuring logical molecular construction.

And then the simplicity is another advantage because it is easier to implement compared to more complex models like GANs and VAEs, yielding reasonable results. And then versatility, because it works with smile strings and selfies, which is another, you know, method to represent the molecules and sometimes the graph-based data as well. However, there are some challenges, such as exposure bias, where the mistakes made early in the sequence can affect the later stages. The limited context is that basic RNNs struggle to capture long-range dependencies compared to, for example, attention-based models in complex molecular structures. So, they sometimes struggle there, and the efficiency suffers because step-by-step generation can be slower than the other generative models.

So, some of the popular RNN models are Reinvent, which is a reinforcement learning-based RNN model designed to create molecules with specific properties, optimizing based on goals like binding affinity or drug likeness using the SMILES strings. So, we can say that Reinvent is a chemical language-based model using reinforcement learning and recurrent neural networks to generate new molecules, and it can also optimize those molecules because it uses reinforcement learning. So, it can optimize those molecules for the end properties as well, such as binding affinity or drug likeness. Then, coming to another model that is extensively used, which is known as variational autoencoders. So, there are probabilistic generative models that encode molecules into a continuous latent space and decode this representation to generate similar molecules.

So, this latent space allows smooth navigation enabling interpolation, optimization and sampling of new molecular structures. So, you can see here that we have the input structure. And then we have an encoder, which is again a neural network, and this encoder generates this latent space, okay. And then we have a decoder; this decoder is able to generate the molecules based on these properties in the latent space, and then we get the output.

So, we have a yes, okay. So, we just talked about it, and then again it can be smile-based VAEs, which use character sequence strings with RNNs or transformers, or they can be graph-based VAEs, which use molecular graphs with graph neural networks. So, some of the advantages of VAEs are latent space exploration. So, it enables the exploration of meaningful molecular variations, and optimization is another advantage in which the latent vectors can be optimized to generate molecules with the desired properties. So, if I want a molecule to be soluble and have a logP between 3 and 5, So, these properties can be used as the latent space variables, and then they can be used as the latent vectors, and then we can use this information to generate the optimal molecules with these two properties. And then interpolation allows for generating molecules by interpolating between the known

molecules as well.

However, there are some challenges, such as the validity of the outputs. So, the decoder may generate invalid or chemically implausible molecules as well, and that is one of the big challenges of VAE-based models and de novo drug design models. Then the mode collapse where the decoder may focus on a small subset of the latent space reducing the diversity. So, in this case, the model can only generate molecules related to some of the input structures. So, that could be one of the issues, and then the training difficulty requires balancing reconstruction loss and KL divergence.

So, these are what we will talk about while discussing the evaluation metrics for de novo generative models. And then, the popular model for these VAEs is a graph VAE, which learns to generate molecular graphs by encoding and decoding adjacency and feature matrices, enabling direct graph-based molecule generation. And then we have generative adversarial networks (GANs). So, GANs are, you know, generative models with two networks: a generator and a discriminator. So, the generator creates new molecules while the discriminator evaluates whether they are valid.

So, the GAN works through a competitive process where the generator improves by trying to fool the discriminator into believing that the generated molecules are real. So, we have a generator and a discriminator, where the generator, as I said, is generating new molecules; it can be based either on chemical language or on graph-based methods as well. From random noise or latent space, we have a discriminator that assesses the authenticity of the generative molecules, distinguishing between real and fake molecules. So, we have smile-based GANs and graph-based GANs, where in the smile-based GANs the molecules are represented as character strings, with the generator learning to produce valid smile sequences. And in the graph-based GANs, as with the other models, molecules are represented as graphs where the generator creates molecular graphs in which the nodes represent atoms and the edges represent bonds.

Some of the advantages of these GANs are their diversity. So, it can generate a wide range of novel molecules by sampling from the latent space and then producing high-quality output by optimizing the generator to fool the discriminator. So, GANs can produce high-quality, realistic molecular structures, and they have quite a lot of generative power. So, it can create new molecular candidates for further optimization and property predictions. However, some challenges exist; one of the challenges is training instability.

So, GANs often face training difficulties, with mode collapse being common, where the generator produces molecules that are similar to the training dataset. And then the validity of the output may produce invalid or chemically infeasible molecules as well. And then,

the evaluation of generated molecules, while assessing whether they are useful, often requires additional filtering and validation. And then, MolGAN is one of the popular models that generate molecular graphs and optimize molecules for the desired properties. So, if we look at the evolution of GANs, a lot of foundational research was done in the beginning, and then GAN development occurred, resulting in many GANs being used in drug discovery in recent years.

Another model is the graph neural network. So, they treat molecules as graphs where the atoms are nodes and the bonds are edges to learn the structural representation for generating new molecules, capturing the relational and spatial information inherent in the molecular structure. So, that is one of the biggest advantages of these GNNs: they can capture information about how the drug is binding to the protein as well. So, they can take that information, and it can be embedded into those models to generate new molecules. So, ah, talking about its architecture. So, we have those node-edge feature initializations in which atoms and bonds are initialized with feature vectors.

And then we have message passing, where each node aggregates information from its neighbors to update its features. And then, the graph readout, where a global molecular embedding is obtained by pooling the node features, is presented. and generation the learned graph representation is decoded to construct the new molecules atom by atom or substructure by the substructure. So, they can take, you know, the molecules represented as graphs, and it supports both small molecules and large chemical structures, or even protein structures as well. And the advantages of the GNNs are the chemical intuition because they naturally capture the molecular topology; they can even take into account the 3D structure and the conformation as well.

It is more relevant when we talk about biologically active molecules because they have only a limited number of conformations that are biologically active and bind to the target in that form. So, flexibility is another advantage; it handles arbitrary molecular sizes and structures, and precision is also an advantage, as it preserves chemical rules. Connectivity is better than the smile-based models, and that is why most of those GNN-based models are, you know, better at producing valid molecules compared to the chemical language models. So, the chemical language model might produce about 60 to 70 percent valid molecules, while GNN-based models can produce about 92 to 95 percent valid molecules when we use them for de novo drug design. However, there are some challenges, such as complex generation: graph generation is non-trivial and may require stepwise construction.

So, scalability is another challenge because it is computationally intensive, especially with large graphs, and there are also validity constraints; we must ensure chemical validity during generation. So, popular models are the graphs invented by AstraZeneca. So, we

know that this molecule is interacting with one of the targets with all those hotspot amino acid residues in the binding pocket. So, now if we know that information, it will be easier for us to take it into account in the de novo generative modeling. So, for that, we use various methods to represent the small molecules, as we have seen earlier as well.

We can represent the smiles, graphs, property grids, the Euclidean distance matrix, and the Cartesian coordinates as well. If we want to incorporate the protein information, we need to do the conditional generation. So, we have the chemical library, and then we have the protein complex. So, now we can take into account this information from the binding pocket, train this model, and generate the molecules that have favorable interactions with this protein. That is one method by which we are using conditional generation.

Another method is docking-based backpropagation, which can also be called reinforcement learning, where we use the docking score as a criterion or scoring method. So, we have the deep generative network, which is generating the molecules; we evaluate the molecules that are generated by it. By using the docking, and then again based on that score, we generate new molecules. So, we have those transformer-based models for de novo drug design.

So, we talked about those transformers earlier, as well. So, these are attention-based deep learning models that excel in sequence modeling. So, in drug design, they are used to generate novel molecules by learning patterns in molecular sequences, such as SMILES or graphs, thereby enabling context-aware and chemically valid generation. So, they have the encoder, decoder or decoder only. So, if we talk about the architecture, it can have the encoder, decoder, or decoder-only setup, which learns the distribution of molecular sequences. And then they have the self-attention mechanism, which is the most important thing, as it can capture long-range dependencies and chemical substructure patterns.

And then they have the positional encoding, which adds order awareness to the input sequence. And then it uses autoregressive generation, where it generates one token at a time, conditioned on previous tokens. So, the input representation in these transformer-based models can be both smile-based and graph-based. So, in the smile-based representation, those molecules are represented as character strings, which are suitable for language modeling. And then the graph is based on, so you know those molecules are being represented as the graphs actually.

And then we usually get some of those tools that are pre-trained on large chemical datasets like Zinc and ChEMBL for transfer learning as well. So, talking about the advantages, scalability is one of the benefits, as we can easily parallelize these models and train them on large data sets. And then chemical grammar learning; it learns the complex syntax and

semantics of the molecules as well, and they are context-aware. So, it can capture dependencies across distant atoms or tokens, and they are quite flexible in generating molecules.

So, they can be conditioned on properties, scaffolds, or targets. So, if we want to generate target-specific molecules, these transformer-based models can be quite useful. However, there are some challenges, such as the fragility of smiles. So small token changes may yield invalid molecules and then use memory because self-attention is computationally expensive for long sequences. So, they need a lot of, you know, computational power, and interpretability is another issue because attention maps may be hard to interpret chemically. And then some of the popular models, like Chemberta, MOL-GPT, or drug X, where, for example, MOL-GPT is a GPT-style autoregressive model generating molecules from scratch.

Okay, and then we have the, you know, reinforcement learning, which is again a kind of machine learning, actually. So, it is a use of fine-tuning generative models, like we can use RNNs or transformers to generate molecules that maximize specific objectives such as bioactivity, drug likeness, or synthetic accessibility. So, in the case that I want to generate a molecule, I need a kind of water-soluble or BBB-permeable molecule. So, what I can do is use this reinforcement learning, where I will work on the basis of this reward and punishment. So, I mean, in that case, we need to use a scoring method that can come from, you know, any method; for example, it can be a QSAR method, or it can be, you know, property prediction, or it can be a pharmacophore-based evaluation, so all that.

In the RL AH models, we have an agent that is a generative model, and usually, these are RNNs or transformers that propose new molecular structures in SMILES format. So, here you can see that the agent is present. So, we have the environment that evaluates the generated molecules and provides a reward based on the desired properties, as we have this agent that is producing new molecules, generating new molecules. And then we have an environment, which is, for example, our scoring method, like docking-based scoring, and the generated molecules will be scored based on this environment. A reward or punishment will be given to this learning algorithm, and it will generate those molecules based on this process only.

So we have the policy where a strategy maps the current state to the next action, and we have the reward function, which quantifies how well a molecule meets the target criteria, such as the docking score, or it can be either QED, which is the quantitative estimate of drug likeness, or it can be toxicity as well. And then the training process starts with a pre-trained generative model; we use it to optimize the molecule to maximize expected rewards. And then we use a feedback loop where the agent samples the molecules, the

molecules are evaluated, and then a reward or a punishment is given; the agent is updated, and then it generates new molecules. So, the biggest advantage of using these RL models in de novo drug design is the goal-directed generation. So, for example, I wanted to generate a molecule, as I said, that is either water-soluble or BBB-permeable.

So, I can use this method because it can learn to generate molecules that specify multiple property constraints. Flexibility is another advantage. Where we can optimize for hard-to-model objectives like the dotting score or the patent novelty. Integration is another advantage in which we can combine well with other generative approaches like VAEs or transformers. However, challenges exist, such as sparse or delayed rewards, and many generated molecules may be invalid or irrelevant.

The exploration-exploitation trade-off balances novel discoveries versus optimizing known good areas, and mode collapse may cause the agent to generate a narrow set of similar molecules repeatedly, which was a limitation with other models as well. So, the popular models are like re-invent, which is an RNN-based smile generator fine-tuned using RL for drug-like properties, and then Mole DQN, which uses Q-learning to modify molecular graphs step by step. Okay, so these were, you know, some of those models, some of those deep generative models that are being extensively used in de novo truck design. There are other models as well, but we could not cover all of them. So, let us talk about some of the challenges in these, you know, deep generative models for truck discovery.

So, one of the biggest challenges is the validity issue because those models may generate molecules that are syntactically correct but chemically invalid or unstable. And then, mode collapse is another issue that we have seen in many of those models, you know, that generators sometimes produce limited structural diversity, failing to explore the full chemical space. And then, rewarding design complexity in RL, a purely designed reward function can lead to non-meaningful or unrealistic molecules. Data bias, scarcity, and a limited and biased molecular data set restrict model generalizability across the drug classes. The lack of interpretability in the decision-making process for deep models is often opaque, making it hard to understand why specific molecules are generated.

And that is, you know, a big challenge because this is an overall challenge for all those deep generative models that we cannot really, you know, quantify the factors which are responsible for the outcome. And then synthetic infeasibility is another big challenge for these de novo generated generators because many generated compounds are not synthesizable using the current chemical methods. And if they cannot be synthesized, they cannot be tested, and they cannot be validated, actually, because without wet lab validation, they will be of no use in generating those molecules. And then multi-objective optimization balancing multiple drug-like properties simultaneously is difficult and often leads to trade-

offs. Evaluation challenges are also present because we do not have any universally accepted metrics, which makes it hard to fairly compare generating model performance.

And then above all, the computational cost is, you know, a big challenge, especially for those graph-based models, because training and deploying those models require significant time and computational resources, which are not, you know, accessible to many researchers in the world. Okay, let's come to the summary. So, deep generative models are revolutionizing drug discovery by enabling de novo molecular design with unprecedented speed, precision, and innovation. And by leveraging architectures such as VAEs, GANs, RNNs, and transformers, these models can generate novel compounds tailored to desired properties. While they are shifting the paradigm from random screening to rational AI-driven design.

So, they address key challenges in early-stage drug development by significantly reducing time, cost, and experimental effort. As AI continues to integrate more deeply with chemistry and biology, deep generative models are poised to transform how we discover, optimize, and personalize future medicines. So, I suggested some of the references that you can go through if you want to learn more about this topic. And then, in the end, I have a small activity for you. So, you shall conduct a literature review and compile a list of at least ten tools based on chemical language models for de novo drug design. And with that, thank you.