IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 13 February 2024, accepted 8 March 2024, date of publication 18 March 2024, date of current version 1 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3377289

RESEARCH ARTICLE

Geom-SAC: Geometric Multi-Discrete Soft Actor Critic With Applications in De Novo Drug Design

AMGAD ABDALLAH[1], NADA ADEL[2], A. M. ELKERDAWY[3,4], SHIHORI TANABE[5], FREDERIC ANDRES[6], (Senior Member, IEEE), ANDREAS PESTER[1], AND HESHAM H. ALI[7]

[1]Faculty of Informatics and Computer Science, AI Group, The British University in Egypt, El Sherouk City, Cairo 11837, Egypt
[2]Faculty of Engineering, Department of Computers, Communications and Autonomous Systems Engineering, New Giza University, First 6th of October, Giza Governorate 3296121, Egypt
[3]Department of Pharmaceutical Chemistry, Faculty of Pharmacy, Cairo University, Cairo 11562, Egypt
[4]School of Pharmacy, College of Health and Science, Joseph Banks Laboratories, University of Lincoln, LN6 7DL Lincoln, U.K.
[5]Division of Risk Assessment, Center for Biological Safety and Research, National Institute of Health Sciences, Kawasaki-ku, Kawasaki 210-9501, Japan
[6]National Institute of Informatics, Chiyoda-ku, Tokyo 101-8430, Japan
[7]College of Information Science and Technology, University of Nebraska at Omaha, Omaha, NE 68182, USA

Corresponding author: Amgad Abdallah (amgad.abdallah@bue.edu.eg)

This work was supported in part by the National Institute of Informatics under FY2023 Strategic Program.

ABSTRACT Finding new molecules with desirable properties has high computational and overhead costs. Much research has focused on generating candidate molecules in one- and two-dimensional spaces, which has produced some favorable results. However, extending these approaches to molecules in three-dimensional space would be far more useful because the representation of molecules is more realistic, although three-dimensional methods have much higher computational costs. In this work, we developed a geometric deep reinforcement learning agent that generates and optimizes molecules that could interact with a biochemical target. The agent can be used for generating molecules from scratch or for lead optimization when it enhances the properties of a given molecule, whether by enhancing its drug-likeness or increasing its activity toward the target via implicit learning. Thus, the agent works with molecules in three-dimensional space without high computational costs.

INDEX TERMS Molecule generation, molecule optimization, geometric deep learning, reinforcement learning, de novo drug design.

I. INTRODUCTION

Searching for new molecules with desired properties is crucial, but the computational overhead and time costs accompanying this process are huge, reaching 3 billion dollars over the past 10 to 15 years [1]. Artificial intelligence (AI) is undergoing a Cambrian explosion in algorithms and has had major successes in solving complex problems. The present work applies AI to the problem of finding the best molecule to satisfy a quantifiable objective, such as obtaining the desired properties. Molecules can be generated in an auto-regressive manner by sequentially creating or editing a molecule via actions such as adding and removing bonds and atoms step-by-step until a final molecule is obtained. Alternatively, molecules can be generated in one shot by learning to generate an entire molecule at once from the latent space. Different methodologies have been developed to find molecules with desired properties. One approach is to build a generative model that can learn the underlying structure of the chemical space and generate novel molecules with desired properties, such as high binding affinity. The model can also maximize the quality of the molecule as a drug candidate by maximizing drug-likeness functions, such as the quantitative estimate of drug-likeness (QED), which

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Wang.

is a computational approach that assesses the likelihood of a molecule being a successful drug candidate based on its chemical properties. The search or optimization process in the latent space can then be guided by these properties to identify promising drug candidates for further testing and development. This approach has shown great promise in accelerating the drug discovery process and has the potential to speed up the introduction of new drugs to the market, which could save lives, money, and time. Molecules have been represented in many ways, including as a one-dimensional (1D) string and most commonly as the simplified molecular-input line-entry system (SMILES), which follow chemical rules to ensure the validity of the molecule. Using this kind of representation, it is intuitive to apply a sequence generative model, such as recurrent neural networks (RNNs). For example, Grisoni et al. used a bidirectional RNN and Kusner et al. sampled from the latent space using a variational autoencoder model that followed encoding and decoding regime [2], [3]. However, string generation methodologies still suffer from severe limitations, such as poor output molecule validity and difficulty optimizing for desired properties. Another representation method is two-dimensional (2D) molecular graphs, in which atoms and bonds are modeled as nodes and edges. This method has gained attention because it can model structures that are more complex. Advances have been made in graph neural networks (GNNs), which have greater expressive and representational power than traditional convolutional neural network architectures [4]. Another promising approach that leveraged feature extraction of molecules was introduced by Xing et al. [5] proposed and validated a method for geometric feature extraction from drug point clouds using DGCNN [6], addressing challenges in existing methods and emphasizing the importance of point cloud data in chemical reactor management. The introduction of DGCNN enhances precision, recall, F1 score, and accuracy, demonstrating its superiority over traditional methods like PointNet [7] and PointNet++ [8]. Furthermore, the studies contribute to the field by creating a comprehensive drug point cloud dataset, filling a gap in data sets for end-to-end extraction models, and advancing the understanding and application of DGCNN in the automated and intelligent processing of drug point cloud data, with potential implications for pharmaceutical process optimization. However, using DGCNN appears to be a more practical solution, it is not without challenges. One challenge is that it depends on a transformation network to adjust the point cloud, yet this introduces a doubling of the network size. Additionally, the deep features and their neighbors might exhibit excessive similarity, making it difficult to generate valuable edge vectors. Furthermore, the DGCNN involves numerous trainable parameters, addressing difficulty in identifying optimal parameters during the comprehensive training of the entire network. Moreover, molecular graphs can be easily extended to model the three-dimensional (3D) structure of molecules, which is a more realistic representation of the molecule that has greater utility for drug design because it considers the conformation of the molecule. The 3D structure also reflects more molecular properties, such as energy or other quantum properties, and is better for predicting binding with proteins or understanding many biological interactions that are useful in drug design [9]. However, working with 3D molecules adds nontrivial computational cost because of the conformer optimization of the molecule, which is difficult as there is no unique solution or infinite solutions. This optimization problem has been tackled by minimizing the energy of the molecule to achieve higher stability, but many computational chemists do not regard energy as relevant to drug design problems.

II. RELATED WORKS

The most relevant works are those that use GNNs or reinforcement learning for molecule generation and optimization tasks, and we analyze the strengths and gaps in these methods.

A. EARLIER WORKS

Zhou et al. introduced MolDQN, in which they optimized molecules with a deep Q-network, and they represented molecules with molecular graphs. They biased the generation of the molecule towards higher QED values by sequentially adding an atom or bond, or by removing a bond for each time step. One downside of this approach is the limitation in scalability as it only generates relatively small molecules [10]. Jin et al. obtained molecules by identifying substructures extracted from existing molecules identified as possibly having favorable properties, and then expanding and fine-tuning molecules using a policy gradient method. Although this approach had good results, it had limited suitability for searching for new molecules [11].

B. MOLECULES IN 3D

Recent approaches focus more on 3D representations of molecules, and thus are more powerful, have more potential capabilities, and can perform more tasks. In the following section, we discuss some of these methods. Joshi et al. relied on the correlation between the functional groups of a molecule and its desired properties instead of generating molecules from scratch. The framework used a method that involves constructing molecules around a pre-determined scaffold by adding individual atoms sequentially within the 3D space of the scaffold. The need for a preexisting scaffold restricts the exploration space and the optimization toward higher-quality drug likeness properties, such as QED, is limited. Nevertheless, the method guarantees that the final molecule always has a desired scaffold [12]. Hoogeboom et al. introduced an equivariant denoising diffusion model for generating molecules in 3D space by using a diffusion process, in which noise is incrementally introduced to both positional data and atom types at discrete time intervals. At each interval, the model outputs a Gaussian distribution with a growing variance. This noise is introduced to the

positional data and atom types and provides both rotational and translational equivariance. The generation process concentrates on mastering the reverse denoising process, in which a neural network is learnt to predict the mean of the Gaussian distribution. The model selects a molecule from the diffused molecules and inputs it into a neural network capable of processing both continuous and discrete data. The method is applicable to both discrete and continuous data, but diffusion-based models have a higher computational cost [13]. Cho et al. [14] proposed a reinforcement learning (RL) approach using a deep deterministic policy gradient algorithm for optimizing the 3D Euclidean geometry of small molecules. They intended to obtain higher-stability molecules by optimizing so that the molecule energy was within ± 3.0 eV of the true total energy of the target molecule. The episode reached an end if the model placed two atoms too near to each other, which broke the chemical bond and caused an error in the density functional theory algorithm; when the model predicted two connected atoms to be so far from each other that the chemical bond broke; or when the molecule was within the specified energy range. This approach assumed that the optimization process was deterministic and developed a smooth technique to handle Euclidean space; however, the work focused on small molecules.

III. PROPOSED APPROACH

In this section, we describe our extension of the soft-actor critic (SAC) algorithm to work with multi-discrete action spaces, and we highlight our contribution to solving the problem of finding new molecules with desired functionalities. Working with molecules in 3D has considerable promise as highlighted previously, yet it is a more complicated task and requires more computational resources than working in 1D or 2D. Therefore, there are extra limitations on our expectations for the output; for example, we need to account for extra features in the modeling added by increasing the dimension, including the torsion, angles, and distances. Reinforcement learning faces some challenges because the environment needed for this model is a mixture of continuous spaces for atom positions and discrete spaces for discrete actions, such as atom and bond addition. We propose a solution for these limitations that focuses on flexibility to allow easy use and tweaking by chemists so that the method meets the need for multi-objective molecular generation and optimization for drug design. Our contributions can be summarized as follows. In order to differentiate our work over existing approaches, Geom-SAC is capable of handling very complex and, high-dimensional tasks with improved sample efficiency, stability, and scalability compared to other RL agents. Moreover, most methods have been based on learning a value function or using on-policy algorithms. Off-policy algorithms are more sample-efficient than on-policy algorithms; thus, the SAC algorithm outperforms other methods in the exploration process because this algorithm is trained to maximize the bargain between the expected return and entropy. Consequently, we propose an extension to the SAC algorithm to allow it to operate on graphs and we use it to handle multi-discrete action spaces. Our approach combines the low computational costs when dealing with 2D molecular graphs and the advantages of the 3D state, delivering a multi-objective tool for chemists to use in the next generation of computer-aided drug design. Molecules are encoded using a hybrid GNN model with a flexible integration between robust, fast, and well-maintained frameworks, such as PyTorch Geometric (PYG), Gymnasium, and RDKit.

A. MDP FORMULATION

First, we developed our environment, in which we define our problem of generating or modifying molecular graphs as a Markov decision process (MDP). The environment allows smooth integration among PYG for graph neural networks and molecular graph modeling; Gymnasium [15], which is a maintained fork of OpenAI's Gym framework that provides tools needed for reinforcement learning; and RDKit, which provides functionalities to perform calculations and operations on molecules. Our MDP is different from other work in several aspects. Usually at the beginning of the episode, invalid actions are taken; for example, the agent may output invalid indices for atoms to be connected with a bond. As an intuitive solution to this problem, we map these illegal actions to legal ones, and we impose a penalty on the agent when it takes such an action. Moreover, we early stop the episode if too many invalid actions are taken. In our **MDP**, we track the evolution of the molecule, and we early stop the evolution process if worse modifications continue to be made. At every step, we give the agent an intermediate reward if the modified molecule does not break the MMFF94 algorithm to make sure that the molecule is embeddable in 3D space, produces a chemically valid molecule, or does not violate the chemical valency [16]. We combine features from both domains (graph theory and chemistry), such as node degree and hybridization type. These features are calculated at each time step for the molecular graphs. We do not directly employ a hybrid action space to produce 3D geometries for the molecule atoms for the following main reasons. First, there are many cases where a reasonable 3D conformer for a molecule is unknown; thus, there is not yet a general rule. Second, we intend to provide chemists with a tool for lead optimization, for introducing a new drug to the market rapidly with minimal costs, and for limiting harmful or fatal side effects in a new drug. Third, classical algorithms for molecular embedding into 3D outperform deep-learning approaches in many scenarios. Fourth, despite the remarkable results obtained by present deep-learning approaches in generating molecular geometries, these approaches lack scalability to larger molecules, and because there is no correct answer to which is the right conformer, there could be an infinite number of solutions. Similar to You et al., [17] we defined our action space to be multi-discrete, and we followed their definition for checking the chemical valency and validity

of the molecule. However, we extended the action space to five joint actions at each time step, and we added more methods for more generalization and to benefit from the utility of 3D structures. For observing the states, we defined our graph as **PYG** to output the **Data** object containing the information about the state as a graph where X is the node features tensor $X \in \mathbb{R}^{N\times F}$ and the tuple represents an edge index in COO format, where $I \in \mathbb{N}^{2\times|E|}$ encodes edge indices, and edge attributes $E \in \mathbb{R}^{|E|\times D}$ where D represents the edge features. The MDP is defined by a tuple, (S, A, P, R, γ),

where

- S: set of states,
- A: set of multi-discrete actions,
- $P(s_{t+1}|s_t, a_t) := P(s_t, a_t, s_{t+1})$: state transition probability of the next state, $s_{t+1} \in S$, given the current state, $s_t \in S$, and a concatenation of multi-discrete actions, $a \in A$,
- $R(s_t, a_t, s_{t+1}, r_t)$: bounded immediate reward on each transition,
- γ: is the discount factor, where $\{\gamma \in \mathbb{R} \mid 0 < \gamma < 1\}$.

1) ACTION SPACE

Our actions generate and optimize molecules, and they are designed to give high flexibility in the molecule's evolution and to ensure the chemical validity of the molecule in each step as well as the generated molecular graph structure. For example, the observed molecule should not have isolated nodes because they affect the calculation of the molecular properties. Actions are defined as the concatenation of multi-discrete actions over multi-categorical distribution. Formally, it can be defined as

$$A_t \in \{a_0, a_1, \ldots, a_{n-1}\}; a_i \in \{0, 1, \ldots, N_{i-1}\}, \quad (1)$$

where each action a_i has N_i different possibilities.

In our environment, it is defined as

$$A_t \in \{a_0, a_1, a_2, a_3, a_4\}, \quad (2)$$

where

- $a_0 \in \{0, 1, ..., \text{no. of allowed atom types} - 1\}$,
- $a_1 \in \{0, 1, ..., \text{max no. of atoms} - 1\}$,
- $a_2 \in \{0, 1, ..., \text{max no. of atoms} - 1\}$,
- $a_3 \in \{0, 1, ..., \text{no. of allowed bond types} - 1\}$,
- $a_4 \in \{0, 1, 2, 3\}$.

2) ADDING AN ATOM

at each step, the agent selects atom to place in the molecular graph from a set containing atoms determined by the user. This action is performed when the number of atoms inserted does not exceed the predetermined number by the user, or when a termination condition is met. This is determined by action a_0.

3) ADDING A BOND

at each step, the agent chooses two of the atom indices to connect between them from a_1 and a_2, and the type of the bond is determined by the fourth action. Bond addition happens when a_4 is equal to 0.

4) ALTERING BOND TYPE

this action selects two atom indices in the molecule to check whether they have a bond. If there is a bond, the bond type is modified to the type specified by the fourth action, and if there is no bond, a new one is created. This action is performed when a_4 is equal to 1.

5) REMOVING A BOND

this action selects two atom indices in the molecule and checks whether there is a bond connecting them from a_1, and a_2. If there is a bond, it removes the bond, and if there is not it does not perform an action. This action is done when a_4 is equal to 2.

6) CONNECTING MOLECULAR FRAGMENTS TOGETHER

This action acts as a heuristic to aid our agent. It ensures that there are no isolated nodes in the graph because they are not chemically valid and cause incorrect calculations of the molecular properties. The action checks if there are isolated fragments that could be connected together by checking for possible binding sites in the fragments to form a single bigger molecule with the highest QED, and ensures that the molecules produced are 100% valid. This action is done when a_4 is equal to 3.

7) REWARD SYSTEM

We designed a reward system to give the agent a reward or penalty after performing each step, so that the evolution process is biased towards favorable functionalities. A final reward is given after the episode is terminated when a stopping condition is met. At each step, the agent calculates the QED for the modified molecule. When the QED is below 0.44, the agent gets a penalty with a magnitude proportional to how close the value is to 0, and when the QED is above 0.44, the agent gets a reward with a magnitude proportional to how near the value is to 1. The reward or penalty is calculated as:

$$r_{qed} = \frac{1.5 - \sqrt{\frac{1}{\text{QED(mol)}+\epsilon}}}{C}, \quad (3)$$

where

C : tunable parameter for step reward scaling,

ϵ : added for computational stability.

The agent gets a small step reward when the molecule is chemically valid and does not break valency or the 3D embedding algorithm and gets a similarity score if a reference molecule is given for biasing molecule evolution toward a certain activity. After the episode is terminated, the agent gets a final reward for final molecule validity, valency, convergence with the MMF94 algorithm (see Fig. 1), and final QED, as well as a similarity score if a reference molecule

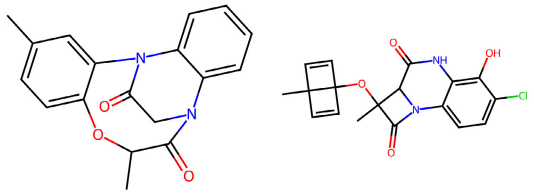

FIGURE 1. Sample molecules that cannot be embedded in 3D space because the MMF94 optimization fails to converge, regardless of the 2D validity or high QED value of the molecule. Our agent implicitly acquires an understanding of the valid 3D geometry for a molecule.

is given with a function similar to the one used in the QED step reward where is the current molecule developed. To ensure that the generated molecule have a distribution similar to that of the reference molecule, we added a step reward for the similarity score by encoding the two molecules into a binary vector using the Morgan fingerprint with radius of 2, after that we calculate the Tanimoto coefficient in order to get the similarity score of the two molecules ([18], [19]).

B. THE GEOM-SAC ALGORITHM

For the agent to learn the policy given the state as a molecular graph, we must encode the state first. Next, the encoded state is given to our agent, which then tries to maximize the trade-off between expected return and entropy using soft policy iterations.

1) GRAPH ENCODER

For node embeddings, we use a hybrid model of a graph attention network (GAT) and a graph isomorphism network (GIN). The attention layer in GAT focuses the algorithm to the most important features of the graph. This combination of GAT and GIN outperforms ordinary graph convolutional networks (GCNs) in terms of generalization to unseen data. GINs have better discriminative powers than GCNs because GINs resemble the Weisfeiler-Lehman test for graph isomorphism, an overview of these methoods and more can be found at [20]. We use the GAT with eight-headed multi-head attention, and then the computed embeddings are passed to the GIN to increase the expressive power with an "add pooling" aggregator as a graph readout [21]. The computed embeddings are defined as

$$h_a = \text{GAT}(X, (I, E)) \tag{4}$$

$$h_\delta = \text{READOUT}(\text{GIN}(h_a, (I, E))) \tag{5}$$

$$h_G = \text{ReLU}(\phi(\text{ReLU}(\omega(h_\delta))) \tag{6}$$

where

- $h_a = [head_1 \| head_2, \ldots, \| head_k]$,
- $X \in \mathbb{R}^{\mathcal{N} \times F}$: node feature tensor,
- $I \in \mathbb{N}^{2 \times |E|}$: tuple of tensors holding edge indices in COO format,
- $E \in \mathbb{R}^{|E| \times D}$: tensor holding edge attributes,
- ϕ and ω: trainable parameters,

$$head_k = \mathbf{W}_i \sum_{i \in \mathcal{N}_j} \beta^k_{i,j} h_j \tag{7}$$

$$\beta_{i,j} = \frac{\exp(a^T [\mathbf{W}h_i \| \mathbf{W}h_j])}{\sum_{j' \in \mathcal{N}_i} \exp(a^T [\mathbf{W}h_i \| \mathbf{W}h'_j])} \tag{8}$$

$$h'_j = \sigma\left(\sum_{j \in \mathcal{N}_i} \beta_{i,j} \cdot \mathbf{W}h_j\right) \tag{9}$$

- $\beta_{i,j}$: denotes the attention on neighbour $j \in \mathcal{N}_i$ when the information is aggregated at node i,
- $\mathbf{a}$: learnable attention tensor of rank one,
- $\mathbf{W}$: learnable attention tensor of rank two,
- $\mathbf{h'_j}$: computed node features,
- $\|$: concatenation operator.

$$GIN(h_a, (I, E)) = MLP^{(l)}((1 + \xi^{(l)}).h_i^{(l-1)} + \sum_{j \in \mathcal{N}_i} h_j^{(l-1)}, \tag{10}$$

- $\xi^{(l)}$: trainable parameter.

In our definition, for the graph encoder's layers we follow arguments and notation as **PYG**.
For the READOUT function, we use a global add pooling [21].

$$h_g = \sum_{i=0}^{N} h_i \tag{11}$$

2) DEEP REINFORCEMENT LEARNING

We choose deep reinforcement learning as our optimization tool for non-differentiable functions because we want to learn both promising and unpromising solutions to our problem. The SAC algorithm is the state-of-the-art algorithm for continuous action spaces, and we are proposing a new version of SAC which can operate in a multi-discrete action space settings, We introduced our problem setup in Section III-A; in the present section, we complete the definition of our algorithm.

a: VALUE FUNCTIONS (V)

Two types of value function are typically used in the **SAC** algorithm for calculating Bellman equations. The first is **state-value function** (V), which estimates the expected cumulative future rewards starting from a given state and following policy π. The second is **twin Q-value functions ($Q1$ and $Q2$)**, which estimate the expected cumulative future rewards starting from a given state, taking a specific action, and then following policy π. This helps mitigate overestimation bias, which is a common problem in Q-learning algorithms.

b: POLICY NETWORK

The policy network, $\pi(a|s)$, resembles the actor part in our algorithm, providing a multi-categorical probability distribution over the multi-discrete actions given the current

state. The objective for the policy is formally defined as

$$J(\pi) = \mathbb{E}[Q(s,a) - \alpha * \mathbb{H}(\pi(\cdot|s))], \tag{12}$$

where α: temperature parameter that controls the trade-off between exploration and exploitation. Higher α emphasizes exploration, whereas lower α emphasizes exploitation.
$\mathbb{H}(\pi(\cdot|s))$: entropy of the policy distribution for state s, which encourages the policy to explore by favoring actions with higher uncertainty. The optimization objective for the actor in actor-critic methods can be defined as

$$\pi^*(a|s) = \underset{\pi}{argmax}\, \pi(a|s)[\mathbb{E}[Q(s,a) - \alpha * \mathbb{H}(\pi(\cdot|s))]]. \tag{13}$$

Here, the term $\alpha * \mathbb{H}(\pi(\cdot|s))$ in acts as a penalty that discourages the policy from becoming overly deterministic, promoting exploration by maintaining a certain level of randomness in action selection.

c: ENTROPY REGULARIZATION

The SAC algorithm encourages exploration by maximizing the entropy, and the entropy regularization term of the policy is defined as

$$\mathbb{H}(\pi(\cdot|s)) = -\sum_{a} \pi(a|s) * log(\pi(a|s)). \tag{14}$$

d: SOFT BELLMAN EQUATION

Describes the relationships between Q-values, state values, and the entropy regularization term. Moreover, this equation provides the foundation for updating the value functions and the policy in the SAC algorithm. Formally, it is defined as follows. Given a state-action pair, (s, a), the soft Bellman equation is

$$Q(s,a) = R(s,a) + \gamma * \mathbb{E}[V(s_{t+1}) - \alpha * \mathbb{H}(\pi(\cdot|s))], \tag{15}$$

where

- γ: discount factor,
- $Q(s,a)$: Q-value for taking action a in state s,
- $R(s,a)$: immediate reward for taking action a in state s,
- $V(s_{t+1})$: value function for the next state s_{t+1}.

e: POLICY UPDATE

Policy π is updated to maximize the expected reward along with an entropy term, which encourages exploration.

f: VALUE FUNCTION UPDATE

Value function V and Q-function $\mathcal{Q}$ are updated using a gradient descent-based method to minimize the mean-squared Bellman error.

g: ENTROPY PARAMETER UPDATE

Temperature parameter α is also updated iteratively to ensure that the entropy of the policy remains within a certain range. The GEOM dataset used in this study is publicly available and can be downloaded online [22]. Code is available for reproducibility at [23].

Algorithm 1 Geometric SAC for Multi-Discrete Actions

Input: an instance of the environment, current state of the molecular graph, number of max steps, number of max atoms, number of max invalid actions ratio, and initial parameters for neural networks V_Ψ, $\overline{V}_\Psi$, Q_{Θ_i}, π_Φ $\forall i \in \{1,2\}$
Output: updated parameters of the neural networks, Ψ_i, Θ_i, Φ $\forall i \in \{1,2\}$
Initialization: replay buffer: D with capacity C and batch size B
Initialization: graph encoder: h_G, finite horizon: M
Initialization: hyperparameters: τ, γ
Initialization: *done* $\in$ {False, True}
Terminal conditions: (if the number of max steps is reached) or (if the number of max invalid actions ratio is reached) or (if the number of max atoms is reached) (see 3.1)
for `i = 0:M-1` **do**
  $done_t \leftarrow$ False
  while not done **do**
    $a_t \sim \pi_\varnothing(\, \|Categorical(a_t^i) \mid h_G(s_t))\ \forall i \in \{1,2,\ldots,n\}$
    $D_{t+1} \leftarrow D_t \| (s_t, a_t, r_t, s_{t+1}, done_t)$
    $V_{\psi_{(t+1)}} := V_{\psi_t} - \lambda_{V_{\psi_t}} \nabla_{\psi_t} V_{\psi_t}(s_t)(V_{\psi_t}(s_t)$
    $-Q_{\theta_t^i}(s_t,a_t) + \log \pi_{\Phi_t}(a_t|s_t))\ \forall i \in \{1,2\}$
    $Q_{\theta^i_{(t+1)}} := Q_{\theta_t^i} - \lambda_{Q_{\theta_t^i}} \nabla_{\theta_t^i} \mathbb{E}_{((s_t,a_t)\sim D)}[(Q_{\theta_t^i}(s_t,a_t)$
    $-\mathcal{R}(s_t,a_t)$
    $+\gamma \sum P(s_{t+1}|s_t,a_t) * \overline{V}_\psi(s_{(t+1)}))^2]\ \forall i \in \{1,2\}$
    $\pi_{\phi_{(t+1)}} := \pi_{\phi_t}$
    $- \lambda_{\pi_{\phi_t}} \nabla_{\phi_t} \mathbb{E}_{(s_t)\sim D}[\pi_{\phi_t}(s_t)^T [\alpha \log(\pi_{\phi_t}(s_t)) - V_{\psi_t}(s_t)]]$
    $\alpha_{t+1} := \alpha_t - \lambda \nabla \pi_{\alpha_t}(s_t)^T [-\alpha(\log(\pi_t(s_t)) + \hat{\mathbb{H}})]$
    $\overline{Q}_{\theta^i_{(t+1)}} := \tau Q_{\theta_t^i} + (1-\tau)\overline{Q}_{\theta_t^i} \forall i \in \{1,2\}$
    $s_t \leftarrow s_{t+1}$
    (terminal condition $\implies$ $done_t \leftarrow$ True) $\wedge$ ($\neg done_t \leftarrow$ False)

Algorithm 1: An extension of the SAC algorithm to handle multi-discrete action spaces and operate on graph data structures. To formally prove that the algorithm terminates, we need to establish a termination condition that ensures the Geom-SAC algorithm ends after a finite number of steps. In the Geom-SAC algorithm, there are two main loops: an outer loop over episodes and an inner loop over steps within each episode. We need to examine both loops.

- Outer loop (M): to prove termination for the outer loop, we specify a finite number, M, of episodes. This means that the algorithm runs for a predetermined number of episodes, and then terminates.
- Inner loop (steps within an episode): The inner loop iterates until the termination condition "done" becomes true, which happens if the number of maximum steps is reached, or if the number of maximum invalid actions ratio is reached, or if the number of maximum atoms is reached. This is a common pattern in reinforcement

TABLE 1. Comparison of the top three molecules from our agent against the state-of-the-art approaches. We obtain a molecule with QED of 0.9484, which is higher than the QED of molecules in the databases, making it a candidate for the highest recorded QED recorded.

Method	QED 1^{st}	2^{nd}	3^{rd}
ZINC (Dataset) [24]	0.948	0.948	0.948
GEOM (Dataset) [22]	0.948	0.948	0.948
JT-VAE [25]	0.925	0.911	0.910
GCPN [17]	**0.948**	0.947	0.946
Elend et al. [26]	0.918	0.898	0.890
Romeo et al. [27]	**0.948**	0.947	0.947
QADD [28]	0.944	0.943	0.927
Geom-SAC	**0.948**	**0.948**	**0.948**

learning algorithms, where an episode terminates when certain criteria are met (e.g., the terminal condition in our approach).

IV. EXPERIMENTS AND RESULTS

We tested our approach against the following three tasks to assess the efficiency of the agent:

- **Task (a)**
  – Goal: Generating a molecule from scratch, in which starting from a carbon atom connected to a random atom with an arbitrary bond, the algorithm should generate a molecule that maximizes desired properties. We optimize for a higher QED, chemical validity, and 3D spatial embedding capability.
  – Results: By tracking the trajectory of molecules' evolution during episodes, our agent learns the aromatic structures, and it generates molecules with higher drug-like properties. Table 1 compares our results with state-of-the-art approaches in this task considering the top three molecules.
- **Task (b)**
  – Goal: Optimizing an existing molecule, in which for any molecule, the algorithm searches for the best possible actions to enhance its functionalities. We use the same multi-objectives as used for task (a).
  – Results: The agent builds a molecule given a sub-molecular structure or a molecular scaffold as a starting point. The agent increases the quality of this structure towards a QED score indicating drug-like properties, from 0.4 to 0.9483 (see Fig. 2).
- **Task (c)**
  – Goal: Optimizing a molecule towards certain activity, in which we give the algorithm a reference molecule with known activity toward a certain disease, and a tunable parameter of how similar the two molecules should be. We assume that similar molecules have similar effects on the same target, and thus the evolution of the molecule should be in the direction of favorable properties. In other words, the modified molecule should not deviate much from a given distribution of a reference molecule.
  – Results: We increase the QED for existing molecules without deviating much from a reference molecule. We use molecules from the GEOM dataset that are active against the SARS-CoV-1 virus, and we increase the mean QED of the molecules from 0.55 to 0.83 with a mean similarity of ∼20%. An illustrative example is shown in Fig. 3.

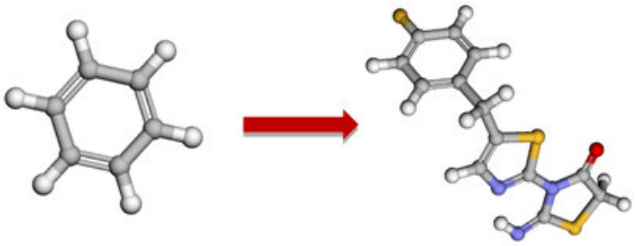

FIGURE 2. Starting from a benzene ring as a scaffold with a QED score of 0.4 (left), the agent raises this value to 0.9483 (right) with minimal modifications of the initial scaffold. The 3D geometry is obtained by the MMF94 algorithm, making it possible to generate conformers and apply quantum mechanics calculations to the molecule. The agent incorporates atom linkage strategies into its reward system to ensure that the 3D geometry of the molecule is valid and the MMF94 optimization converges during the learning process.

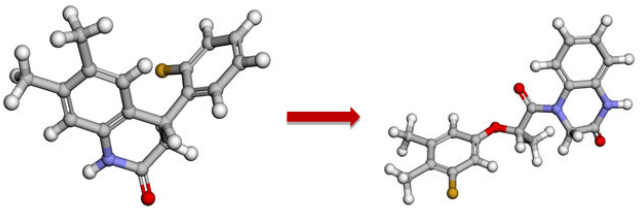

FIGURE 3. A molecule active towards SARS-CoV-1 virus from the GEOM dataset, the QED score of which is increased from 0.83 (left) to 0.93 with a similarity score of 42% (right).

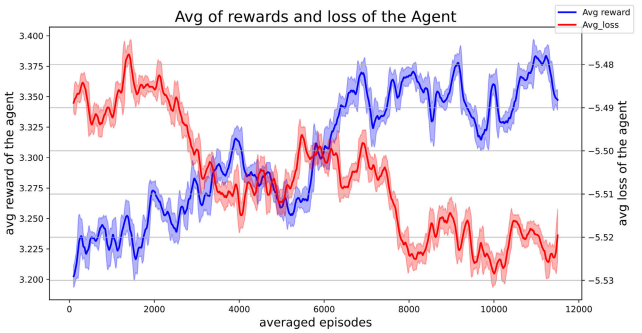

FIGURE 4. Averaged reward vs averaged loss of the agent across all tasks.

To demonstrate the capabilities of our agent, we selected the state-of-the-art approaches as the baseline. Moreover, our model may be generalized so that it may be used not only in drug discovery, but also in similar tasks, such as material design. In drug discovery, we showed that our model outperformed other approaches with the extra advantage of considering geometrical properties in the multi-objective optimization task, in addition to chemical functionalities. The model performance is shown in Fig. 4. Hyperparameter tuning is an expensive task in deep reinforcement learning because the training must be repeated for the same set of hyperparameters many times to detect if there is a substantial improvement in the performance. Certain hyperparameters,

TABLE 2. Best hyperparameters for reducibility.

Hyperparameter	Optimal Value
Attention heads	8
Hidden units' dimensions	128
Batch size	16
Buffer size	100000
Learning rate	0.0003
Optimizer	Adam
Discount factor (γ)	0.99
Soft update coefficient (τ)	0.005
Reward scale	10

TABLE 3. Supercomputer specifications.

Property	Value
Architecture	x86_64
CPU op-mode(s)	32-bit, 64-bit
CPU(s)	8
RAM	62GB

such the number of attention heads, batch size, buffer size, and the dimensions for each hidden layer, were manually adjusted by us, and other hyperparameters were borrowed from the original SAC methodology. Table 2 shows the best hyperparameters for reducibility.

The Magi machine (Table 3) is a high-performance computer with a versatile x86-64 architecture. It can efficiently handle a wide range of tasks, including both older and modern software. It has eight powerful processing units, making it useful for tasks that require multitasking, such as scientific simulations, data analysis, and virtualization. It also has 62 GB of RAM, which is useful for our research. In summary, this machine is a powerful and versatile system that can handle various computing challenges, including intensive calculations, data analysis, and running virtual environments.

V. DISCUSSION

Our algorithm obtained molecules with high QED values. From an initial carbon atom or a molecular substructure, the algorithm learned the carbon ring structure, the atom and bond type distribution given the current state of the molecule, and linking strategies that did not violate valency, chemical validity, or embeddability in 3D space (see Fig. 5).

A. EFFECT OF GRAPH ENCODER ON THE AGENT'S PERFORMANCE

Based on the results of our experiments, we tried many GNN architectures many times, and differences in architecture changed the efficiency of our agent in terms of learning speed and performance. First, we used Dimenet++ because it showed outstanding performance in predicting molecular properties; however, it was too slow, and thus was infeasible in our solution [29]. We also tried a GCN, GIN, and GAT, but the most efficient architecture was a hybrid model consisting of a GAT with eight multi-attention heads and a GIN, and our hybrid model showed superior capabilities even before conducting any type of learning [30].

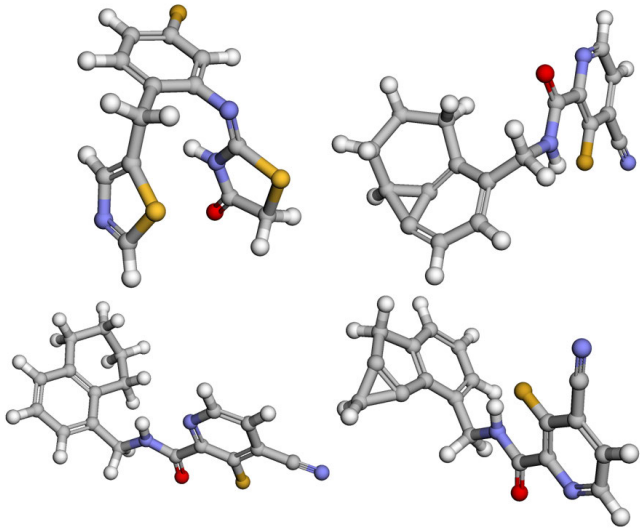

FIGURE 5. Top four molecules generated by Geom-SAC with QED of 0.948 embedded in 3D space via the MMFF94 algorithm.

B. SCALABILITY TO A LARGER AND MORE COMPLEX MOLECULAR STRUCTURES

Our approach is not limited to a certain size of molecules to generate or optimize. In fact, we designed the **MDP** to be very flexible for scalability, for example the user can modify a parameter for setting the maximum size of the molecule before training according to his needs, or available computational/time resources. Moreover, the reward function can be easily modified to bias the generation towards larger molecules by giving small reward proportional to the time steps taken by the molecule, or a final reward proportional to number of atoms in the obtained molecule. However, the complexity of the molecule increases with the increasing the size, specially if the molecule is designed directly in the 3D space. Our approach is designed naturally to handle this complexity of learning the 3D coordinates of the molecule at low cost (see section III-A).

C. MORE COMPARATIVE ANALYSIS WITH OTHER EXISTING METHODS

In terms of computational efficiency, the most expensive step is the molecular graph encoder, which can be approximated by the complexity of GAT and GIN. For GAT, time complexity for computing $h_a^{(t+1)}$ features embedding using a single GAT attention head is $O(|V|h_a^{(t)}h_a^{(t+1)} + |E|h_a^{(t+1)})$, where:

- h_a denotes the number of input features,
- $|V|$ represents the number of nodes in the graph, and
- $|E|$ signifies the number of edges in the graph.

While the time complexity for GIN is $O(|E|)$ if the adjacency matrix is sparse, and $O(|V|^2)$ otherwise. Moreover, Geom-SAC is built on soft actor critic approach which is model-free, online, off-policy, actor-critic reinforcement learning method known to very sample efficiency. In other words, it can reduce the computational/time cost if the process of generating data

is expensive. Regarding the accuracy in real world scenarios, our model is trained to maximize the entropy in its objective optimization function. This implies a notion of creativity in designing or optimizing molecules. In contrast to other approaches that use transfer learning ([31], [32]), it may suffer from limited creativity due to the pre-trained layers.

VI. CONCLUSION AND FUTURE WORK

We provided an extension to the SAC algorithm to handle multi-discrete action spaces, and we used this model to tackle the problem of searching for new drugs. There are still many challenges in computer-aided drug design, such as the need for a better scoring function to assess drug quality or to fill the gaps in the existing functions. For example, although QED is widely used, it has limits to the accuracy with which it gauges the quality of molecules, even when extended to cases involving invalid molecules marked by isolated nodes. Furthermore, the efficacy of computational methodologies is constrained by dataset limitations encompassing inaccuracies, errors, inconsistencies in molecular frameworks, property imbalances, and related factors ([30], [33], [34], [35]). Despite these challenges affecting the accuracy of the proposed computational approaches, research efforts continue to advance this field, yielding incremental improvements regularly. Our approach offers significant economic benefits for the pharmaceutical industry, by efficiently generating and optimizing molecules for drug discovery, Geom-SAC reduces development costs through early identification of promising drug candidates and accelerated timelines. This optimization leads to more effective resource allocation, as pharmaceutical companies can focus resources on molecules with the highest likelihood of success. Additionally, Geom-SAC increases success rates in preclinical and clinical trials by tailoring molecular structures to specific drug targets and properties. These combined benefits result in cost savings, faster time-to-market, and improved competitiveness, enhancing the overall efficiency and productivity of the pharmaceutical industry.

In future work, we may use a hybrid action space for multi-discrete and continuous actions to generate 3D geometries from scratch. Multi-modal data integration will enhance the drug discovery process, because using diverse sources of data, such as genomics, proteomics, and metabolomics data, is crucial. Moreover, the demand for personalized medicine is arising because it promises higher efficiency than general drugs.

ACKNOWLEDGMENT

The authors would like to thank the National Institute of Informatics for its support for this research, all the experts and the three anonymous reviewers for their insightful suggestions and careful reading of the manuscript. The calculations were performed using HPC resources from MAGI, the HPC cluster of Université Sorbonne Paris Nord. The authors thank N. Greneche for his support on MAGI 1. https://wwwmagi.univ-paris13.fr.

REFERENCES

[1] J. A. DiMasi, H. G. Grabowski, and R. W. Hansen, "Innovation in the pharmaceutical industry: New estimates of R&D costs," *J. Health Econ.*, vol. 47, pp. 20–33, May 2016.

[2] F. Grisoni, M. Moret, R. Lingwood, and G. Schneider, "Bidirectional molecule generation with recurrent neural networks," *J. Chem. Inf. Model.*, vol. 60, no. 3, pp. 1175–1183, Mar. 2020, doi: 10.1021/acs.jcim.9b00943.

[3] M. J. Kusner, B. Paige, and J. M. Hernández-Lobato, "Grammar variational autoencoder," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 1945–1954. [Online]. Available: https://dl.acm.org/doi/10.5555/3305381.3305582

[4] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. 34th Int. Conf. Mach. Learn.*, in Proceedings of Machine Learning Research, vol. 70, D. Precup and Y. W. Teh, Eds., Aug. 2017, pp. 1263–1272.

[5] Z. Xing, S. Zhao, W. Guo, and X. Guo, "Geometric feature extraction of point cloud of chemical reactor based on dynamic graph convolution neural network," *ACS Omega*, vol. 6, no. 33, pp. 21410–21424, Aug. 2021.

[6] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, Oct. 2019, doi: 10.1145/3326362.

[7] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 77–85, doi: 10.1109/CVPR.2017.16.

[8] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.* Red Hook, NY, USA: Curran Associates, 2017, pp. 5105–5114.

[9] H. Stärk, D. Beaini, G. Corso, P. Tossou, C. Dallago, S. Günnemann, and P. Lió, "3D infomax improves GNNs for molecular property prediction," in *Proc. 39th Int. Conf. Mach. Learn.*, in Proceedings of Machine Learning Research, vol. 162, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., Jul. 2022, pp. 20479–20502.

[10] Z. Zhou, S. Kearnes, L. Li, R. N. Zare, and P. Riley, "Optimization of molecules via deep reinforcement learning," *Sci. Rep.*, vol. 9, no. 1, p. 10752, Jul. 2019, doi: 10.1038/s41598-019-47148-x.

[11] W. Jin, R. Barzilay, and T. Jaakkola, "Multi-objective molecule generation using interpretable substructures," in *Proc. 37th Int. Conf. Mach. Learn. (ICML)*, vol. 119, 2020, pp. 4849–4859. [Online]. Available: https://dl.acm.org/doi/10.5555/3524938.3525388

[12] R. P. Joshi, N. W. A. Gebauer, M. Bontha, M. Khazaieli, R. M. James, J. B. Brown, and N. Kumar, "3D-scaffold: A deep learning framework to generate 3D coordinates of drug-like molecules with desired scaffolds," *J. Phys. Chem. B*, vol. 125, no. 44, pp. 12166–12176, Nov. 2021, doi: 10.1021/acs.jpcb.1c06437.

[13] E. Hoogeboom, V. Garcia Satorras, C. Vignac, and M. Welling, "Equivariant diffusion for molecule generation in 3D," in *Proc. 39th Int. Conf. Mach. Learn.*, in Proceedings of Machine Learning Research, vol. 162, 2022, pp. 8867–8887.

[14] Y. Cho, S. Kim, P. P. Li, M. P. Surh, T. Y.-J. Han, and J. Choo, "Physics-guided reinforcement learning for 3D molecular structures," in *Proc. 33rd Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2019, pp. 1–6.

[15] M. Towers, J. K. Terry, A. Kwiatkowski, J. U. Balis, G. de Cola, T. Deleu, M. Goulão, A. Kallinteris, A. Kg, M. Krimmel, R. Perez-Vicente, A. Pierré, S. Schulhoff, J. J. Tai, A. T. J. Shen, and O. G. Younis, "Gymnasium," Mar. 2023, doi: 10.5281/zenodo/8148697.

[16] T. A. Halgren, "MMFF VII. Characterization of MMFF94, MMFF94s, and other widely available force fields for conformational energies and for intermolecular-interaction energies and geometries," *J. Comput. Chem.*, vol. 20, no. 7, pp. 730–748, May 1999.

[17] J. You, B. Liu, Z. Ying, V. S. Pande, and J. Leskovec, "Graph convolutional policy network for goal-directed molecular graph generation," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2018, pp. 6412–6422.

[18] H. L. Morgan, "The generation of a unique machine description for chemical structures—A technique developed at chemical abstracts service," *J. Chem. Documentation*, vol. 5, no. 2, pp. 107–113, May 1965.

[19] Y. H. Chung, "A computer program for classifying plants," *Korean J. Plant Taxonomy*, vol. 1, no. 1, pp. 37–43, Jun. 1969.

[20] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, Jan. 2020, doi: 10.1016/j.aiopen.2021.01.001.

[21] D. Grattarola, D. Zambon, F. M. Bianchi, and C. Alippi, "Understanding pooling in graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 2, pp. 2708–2718, Feb. 2024, doi: 10.1109/TNNLS.2022.3190922.
[22] S. Axelrod and R. Gómez-Bombarelli, "GEOM, energy-annotated molecular conformations for property prediction and molecular generation," *Sci. Data*, vol. 9, no. 1, p. 185, Apr. 2022, doi: 10.1038/s41597-022-01288-4.
[23] A. Mahmoud, "Geom-SAC: Geometric multi-discrete soft actor critic with applications in de novo drug design," *Zenodo*, Nov. 12, 2023, doi: 10.5281/zenodo.10116577.
[24] J. J. Irwin, T. Sterling, M. M. Mysinger, E. S. Bolstad, and R. G. Coleman, "ZINC: A free tool to discover chemistry for biology," *J. Chem. Inf. Model.*, vol. 52, no. 7, pp. 1757–1768, Jul. 2012.
[25] W. Jin, R. Barzilay, and T. Jaakkola, "Junction tree variational autoencoder for molecular graph generation," in *Proc. 35th Int. Conf. Mach. Learn.*, in Proceedings of Machine Learning Research, vol. 80, J. Dy and A. Krause, Eds., Jul. 2018, pp. 2323–2332.
[26] L. Elend, L. Jacobsen, T. Cofala, J. Prellberg, T. Teusch, O. Kramer, and I. A. Solov'yov, "Design of SARS-CoV-2 main protease inhibitors using artificial intelligence and molecular dynamic simulations," *Molecules*, vol. 27, no. 13, p. 4020, Jun. 2022.
[27] S. R. Atance, J. V. Diez, O. Engkvist, S. Olsson, and R. Mercado, "De novo drug design using reinforcement learning with graph-based deep generative models," *J. Chem. Inf. Model.*, vol. 62, no. 20, pp. 4863–4872, Oct. 2022, doi: 10.1021/acs.jcim.2c00838.
[28] Y. Fang, X. Pan, and H.-B. Shen, "De novodrug design by iterative multi-objective deep reinforcement learning with graph-based molecular quality assessment," *Bioinformatics*, vol. 39, no. 4, Apr. 2023, Art. no. btad157.
[29] J. Gasteiger, S. Giri, J. T. Margraf, and S. Günnemann, "Fast and uncertainty-aware directional message passing for non-equilibrium molecules," in *Proc. Mach. Learn. Molecules Workshop*, 2020.
[30] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017.
[31] L. P. Queiroz et al., "Transfer learning approach to develop natural molecules with specific flavor requirements," *Ind. Eng. Chem. Res.*, vol. 62, no. 23, pp. 9062–9076, Jun. 2023.
[32] M. Korshunova, N. Huang, S. Capuzzi, D. S. Radchenko, O. Savych, Y. S. Moroz, C. I. Wells, T. M. Willson, A. Tropsha, and O. Isayev, "Generative and reinforcement learning approaches for the automated de novo design of bioactive compounds," *Commun. Chem.*, vol. 5, no. 1, p. 129, Oct. 2022, doi: 10.1038/s42004-022-00733-0.
[33] N. Schneider, D. Lowe, R. Sayle, M. Tarselli, and G. Landrum, "Big data from pharmaceutical patents: A computational analysis of medicinal chemists' bread and butter," *J. Medicinal Chem.*, vol. 59, Mar. 2016.
[34] S. Riniker and G. A. Landrum, "Open-source platform to benchmark fingerprints for ligand-based virtual screening," *J. Cheminformatics*, vol. 5, no. 26, p. 17, May 2013, doi: 10.1186/1758-2946-5-26.
[35] D. Fourches, E. Muratov, and A. Tropsha, "Trust, but verify: On the importance of chemical structure curation in cheminformatics and QSAR modeling research," *J. Chem. Inf. Model.*, vol. 50, no. 7, pp. 1189–1204, Jul. 2010.

AMGAD ABDALLAH received the M.Sc. degree in data science from Cairo University, with a focus on the integration of graph neural networks and deep reinforcement learning in generative AI for drug design.

He is currently an Assistant Lecturer with The British University in Egypt, where he transfers knowledge and experience to his students. Moreover, he is dedicated to advancing pharmaceutical research by employing data-driven methodologies, including computational techniques and machine learning, to expedite the identification and development of new drugs. Also, his focus extends to biomedical informatics, where he explores the integration of mathematical modeling and domain knowledge for developing predictive models for treatment response and adverse reactions, and contributing to the advancement of therapeutic interventions and positively impacting individuals' health. His research interests include applying reinforcement learning, graph neural networks, generative AI, and quantum machine learning.

NADA ADEL is actively engaged in exploring innovative applications of artificial intelligence in the field of chemistry and biology, with a specific focus on understanding molecular structures and interactions. Through her research projects, she has demonstrated a strong competence in applying deep learning techniques to analyze complex chemical data and extract meaningful insights. She is particularly investigating the potential of chemoinformatics to revolutionize drug discovery and molecular design. She actively seeks opportunities to contribute to cutting-edge research initiatives and collaborative projects at the intersection of computer science and chemistry. With a passion for solving real-world challenges via applying deep learning approaches.

A. M. ELKERDAWY received the Ph.D. degree in chemistry from the Computer-Chemistry-Center (CCC) Research Group, Erlangen–Nürnberg University, with a focus on computer-aided drug design (CADD). He is currently an Associate Professor of pharmaceutical chemistry with the Faculty of Pharmacy, Cairo University. He has a unique blend of experiences in synthetic chemistry and CADD. He has published more than 50 articles dealing with using computer-aided-drug design approaches for lead discovery and lead optimization for new targets for the treatment of serious health problems, such as cancer, Alzheimer's, and inflammation.

SHIHORI TANABE received the Ph.D. degree in pharmaceutical sciences from The University of Tokyo, Japan, in 2005, with a focus on molecular biology and pharmacology. She is currently a Senior Researcher with the Division of Risk Assessment, Center for Biological Safety and Research, National Institute of Health Sciences (NIHS). She also performed her research mainly focused on biochemistry and cellular biology with the College of Medicine, University of Illinois at Chicago. A part of her research is focused on gene expression profiling and finding important factors of various cells, including cancer cells and stem cells using a bioinformatics approach including microarray and real-time RT-PCR techniques. She has also conducted research, such as cellular signaling, cardiac cell research, and protein regulation mechanisms. She made contributions in the fields of molecular biology, pharmacology, and biochemistry. Her current research interests include analyzing molecular networks and profiling cancer cells and stem cells in terms of cellular phenotype and cellular signaling. She is also interested in finding essential factors to distinguish cancer cell types for revealing the mechanism of cancer development and safe application of the cells as cellular therapeutics in regenerative medicine, as well as in assessing the long-term influences of the alteration in gene expression and cellular phenotype from the point of view of regulatory science. She has been a delegate for Japanese Society for Regenerative Medicine, a Councilor for the Society for Regulatory Science of Medical Products, a delegate for the Pharmaceutical Society of Japan, and an International Associate for Japanese Pharmacological Society.

FREDERIC ANDRES (Senior Member, IEEE) received the Ph.D. degree in information systems from the University of Paris VI "Pierre et Marie Curie," in 1993, and the Habilitation degree in informatics from the University of Nantes, in 2000. Since 2000, he has been an Associate Professor with the Digital Content and Media Sciences Research Division, National Institute of Informatics (NII), Tokyo, Japan. He is the author of more than 150 papers in international journals, books, and conferences. His main research interests include several topics, such as collective intelligence-based knowledge, AI ethics, data science, supply chain, and pedagogy and didactical web-based science education for all. He was a recipient of the First Prize in the IEEE Brain Data Bank Challenges and Competitions at COMPSAC 2018, in July 2018. He also received the IEEE CertifAIEd Authorized Lead Assessor recognition from the IEEE SA, in September 2023. In addition, he was named "R10 Ethics Champion" by the IEEE Region 10, in October 2023. Since 2018, he has been co-organizing the International DECOR Workshop in conjunction with IEEE ICDE.

ANDREAS PESTER is currently a Professor with the Faculty of Informatics and Computer Science, The British University in Egypt. He is also a Mathematician and a Data Scientist. He has more than 30 years of experience in teaching math and mathematical modeling, simulation technologies, online labs, and machine learning. He was previously responsible for the development of data science and deep learning research and teaching at CUAS, Austria, and BUE, Egypt. In addition to his extensive scientific publications, he was involved in more than 20 EU- and national projects in Austria and Egypt. He is a Reviewer of Ph.D. promotions at UPC Barcelona, UNED Madrid, and the University of Reading, U.K. He was a Guest Professor with the UPC Barcelona; the Technical University of Kharkiv; the Technical University of Kyiv; the University of Maribor; UNESP Bauru, Brazil; the University of Applied Sciences Vienna, Armenian–American University in Yerevan, the Pontifical Catholic University of Rio de Janeiro, and the University of Banja Luka.

HESHAM H. ALI is currently a Professor of computer science and the Director of the University of Nebraska Omaha (UNO) Bioinformatics Core Facility. He served as the Lee and Wilma Seemann Distinguished Dean of the College of Information Science and Technology, UNO, from 2006 to 2021. He has published numerous articles in various IT areas, including scheduling, distributed systems, data analytics, wireless networks, and bioinformatics. He has also published two books in scheduling and graph algorithms, and several book chapters in bioinformatics. He has been serving as the PI or Co-PI of several projects funded by NSF, NIH, and Nebraska Research Initiative in the areas of data analytics, wireless networks, and bioinformatics. He has also been leading a research group that focuses on developing innovative computational approaches to model complex biomedical systems and analyze big bioinformatics data. The research group is currently developing several next-generation big data analytics tools for analyzing large heterogeneous biological and health data associated with various biomedical research areas, particularly projects associated with infectious diseases, microbiome studies, early childhood development, and aging research. He has also been leading two projects for developing secure and energy-aware wireless infrastructure to address tracking and monitoring problems in medical environments, particularly to study mobility profiling for advancing personalized healthcare. He has led several local and national outreach initiatives, including Women in IT initiatives, IT education and training programs, and IT summer internship camps.