Molecular De Novo Design through Transformer-based Reinforcement Learning

Anonymous authors
Paper under double-blind review

Abstract

In this work, we introduce a method to fine-tune a Transformer-based generative model for molecular de novo design. Leveraging the superior sequence learning capacity of Transformers over Recurrent Neural Networks (RNNs), our model can generate molecular structures with desired properties effectively. In contrast to the traditional RNN-based models, our proposed method exhibits superior performance in generating compounds predicted to be active against various biological targets, capturing long-term dependencies in the molecular structure sequence. The model's efficacy is demonstrated across numerous tasks, including generating analogues to a query structure and producing compounds with particular attributes, outperforming the baseline RNN-based methods. Our approach can be used for scaffold hopping, library expansion starting from a single molecule, and generating compounds with high predicted activity against biological targets.

Introduction

The vast expanse of chemical space, encompassing an order of magnitude from $10^{60} - 10^{100}$ possible synthetically feasible molecules (Engelmore & Morgan, 1986), presents formidable obstacles to drug discovery endeavors. In this colossal landscape, the task of pinpointing a molecule that simultaneously meets the prerequisites for bioactivity, drug metabolism and pharmacokinetic (DMPK) profile, and synthetic accessibility becomes an undertaking similar to the proverbial search for a needle in a haystack. Pioneering de novo design algorithms (Clancey, 1983) have attempted to address this by employing virtual strategies to design and evaluate molecules, thereby condensing the vast chemical space into a more navigable realm for exploration.

Traditional de novo design models, based on Recurrent Neural Networks (RNNs), have proven effective in molecule generation tasks (Clancey, 1984; Robinson, 1980a). However, RNNs possess inherent architectural limitations, notably in their capability to capture long-term dependencies in sequential data, which can be particularly detrimental when modeling complex molecular structures. Recently, the Transformer architecture has emerged as a powerful alternative to RNNs in sequence modeling tasks across various domains. Some of the key advantages of Transformers over RNNs include:

1. **Parallelization:** Unlike RNNs which process sequences step-by-step, Transformers process all tokens in the sequence simultaneously, allowing for better computational efficiency.
2. **Long-term Dependency Handling:** Transformers utilize multi-head self-attention mechanisms which can capture long-range interactions in the data, making them particularly well-suited for modeling intricate molecular structures.
3. **Scalability:** Transformers are inherently more scalable, allowing for the processing of longer sequences which is a considerable advantage in molecular design.

In light of these advantages, our work introduces a novel approach by integrating the Transformer architecture, specifically the Decision Transformer, for molecular de novo design. By leveraging the inherent strengths of Transformers, our model exhibits enhanced performance in generating molecular structures with desired attributes.

Furthermore, we emphasize the incorporation of the "oracle feedback reinforcement learning" method. Pretraining models on large datasets is beneficial, but downstream tasks often require fine-tuning on specific objectives. By integrating feedback from an oracle during the reinforcement learning phase, our approach can efficiently navigate the solution space, optimizing towards molecules with high predicted activity. Such oracle-guided optimization provides an added layer of precision, facilitating the generation of molecules that not only conform to structural constraints but also exhibit high bioactivity, thereby increasing the potential success rate in drug discovery endeavors.

Drawing inspiration from previous work that employed RNNs and reinforcement learning for molecular optimization (Clancey, 1983), our approach distinguishes itself by the adoption and fine-tuning of the Transformer architecture, ensuring superior handling of long-sequence data and paving the way for innovative breakthroughs in the realm of molecular design.

In summary, this work presents a fresh perspective on molecular de novo design, underscoring the potential of Transformer-based architectures, complemented by oracle feedback reinforcement learning, to revolutionize drug discovery methodologies. We envision that our approach will not only set a new benchmark in molecular generation tasks but will also inspire future research in leveraging advanced machine learning architectures for complex scientific challenges.

Related Works

Early de novo design algorithms were structure-based, aiming to grow ligands to fit the binding pocket of the target (Robinson, 1980b; Hasling et al., 1984). However, these methods often generated molecules with poor DMPK properties and could be synthetically intractable. Ligand-based approaches were introduced to create a vast virtual library of chemical structures and then searched with a scoring function (Hasling et al., 1983; Rice, 1986).

Recently, generative models such as RNN-based methods have been used for de novo design of molecules (Clancey, 1979; 2021). They have shown success in tasks like learning the underlying probability distribution over a large set of chemical structures, reducing the search over chemical space to only molecules seen as reasonable. Further fine-tuning of the models was done using reinforcement learning (RL) (Bouville, 2008), which showed considerable improvement over the initial model.

Despite these advancements, challenges such as capturing long-term dependencies in the sequence data persist. The Transformer architecture (NASA, 2015), known for its self-attention mechanism and ability to handle long sequences, has been highly successful in several sequence prediction tasks across domains. Motivated by these successes, we propose the use of Transformer-based architectures in place of RNNs for molecular de novo design.

Molecular assembly strategies, such as string-based approaches like SMILES and SELFIES (Weininger, 1988; Krenn et al., 2020), provide an efficient representation of molecules. Graph-based methods offer an intuitive two-dimensional representation of molecular structures, with nodes and edges representing atoms and bonds, respectively (Zhou et al., 2019; Jin et al., 2018). Synthesis-based strategies, on the other hand, aim to generate only synthesizable molecules, ensuring that the design aligns with real-world applications (Bradshaw et al., 2020; Gao et al., 2022).

Various optimization algorithms have been utilized for molecular design. Genetic Algorithms (GAs) mimic natural evolutionary processes and have been applied in the context of molecule generation using both SMILES and SELFIES representations (Brown et al., 2019; Nigam et al., 2021). Bayesian optimization (BO) is another class of method that builds a surrogate for the objective function, with applications such as BOSS and ChemBO in the molecular domain (Moss et al., 2020; Korovina et al., 2020). Variational autoencoders (VAEs) offer a generative approach, mapping molecules to and from a latent space, with notable methods including SMILES-VAE and JT-VAE (Gómez-Bombarelli et al., 2018; Jin et al., 2018). Reinforcement Learning (RL) techniques, like REINVENT, have also been applied to tune models for molecule generation Olivecrona et al. (2017).

Furthermore, recent advancements in gradient ascent methods, such as Pasithea and Differentiable scaffolding tree (DST), have leveraged gradient-based optimization for molecular design (Shen et al., 2021; Fu et al., 2022).

In light of these developments, our approach integrates the benefits of the Transformer architecture with advanced reinforcement learning techniques, aiming to address the challenges present in current molecular de novo design methodologies.

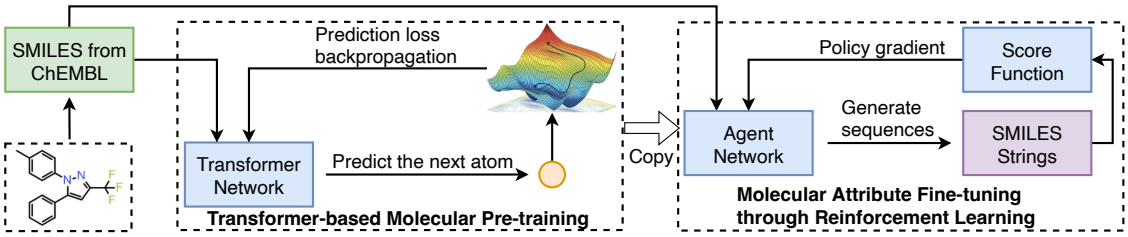

Figure 1: The framework of our method.

Methodology

Our method first pre-trains the real 2D molecule dataset based on the transformer. Then, based on the RL paradigm, fine-tuning is performed on the molecular attributes to be optimized.

Transformer-based Molecular Pre-training

The transformer is used for pre-training on real 2D molecules. Specifically, it treats the prediction of a 2D molecule as a sequence prediction and lets the transformer predict the next atom based on the molecular sequence history. The pre-training of the transformer is based on maximum likelihood.

Transformers Overview Transformers are a neural network architecture designed to process sequential data, while also accounting for the importance of each input in relation to the others, despite their position in the sequence (Goodfellow et al., 2016). They manage to do this by the introduction of an attention mechanism that assesses the significance of each input in the sequence (Figure 1). At any given step t, the transformer state at t is influenced by all previous inputs $x^1, \ldots, x^{t-1}$ and the current input x^t. The transformer's ability to selectively focus on the parts of the input sequence that are most relevant for each step makes them especially well suited for tasks in the field of natural language processing. Sequences of words can be encoded into one-hot vectors with a length equivalent to our vocabulary size X. We may add two extra tokens, GO and EOS, to signify the beginning and end of a sequence, respectively.

Learning the data Training a Transformer for sequence modeling typically involves using maximum likelihood estimation to predict the next token x^t in the target sequence, given tokens from the previous steps (Figure 1). The model generates a probability distribution at every step, representing the likely next character, and the objective is to maximize the likelihood assigned to the correct token:

$$J(\Theta) = -\sum_{t=1}^{T} \log P\left(x^t \mid x^{t-1}, \ldots, x^1\right) \tag{1}$$

The cost function $J(\Theta)$, often applied to a subset of all training examples known as a batch, is minimized with respect to the network parameters Θ. Given a predicted $\log$ likelihood $\log P$ of the target at step t, the gradient

of the prediction with respect to Θ is used to update Θ. This method of fitting a neural network is called backpropagation. Changing the network parameters affects not only the immediate output at time t, but also influences the information flow into subsequent transformer states. This effect does not lead to problems of exploding or vanishing gradients due to the lack of recurrent connections in Transformers.

Generating new samples Once a Transformer has been trained on target sequences, it can be used to generate new sequences that adhere to the conditional probability distributions learned from the training set. The first input is the GO token, and at every timestep following, we sample an output token x^t from the predicted probability distribution $P\left(X^t\right)$ over our vocabulary X. The sampled x^t is then used as our next input. The sequence is considered finished once the EOS token is sampled .

Segmentation and Binary Coding of SMILES

A Simplified Molecular Input Line Entry System (SMILES) (Weininger, 2017) defines a molecule as a character sequence reflecting atoms as well as special symbols that illustrate ring opening and closure along with branching. In the majority of scenarios, SMILES are tokenized

on a single-character basis, with the exception of two-character atom types such as " Cl " and "Br", and unique environments indicated by square brackets (e.g., $[\mathrm{nH}]$), where they are processed as a single token. This approach to tokenization led to the identification of 86 tokens in the training data. Figure 3 illustrates how a chemical structure is converted to SMILES and binary-coded representations.

A single molecule can be represented in multiple ways using SMILES. Algorithms that consistently represent a particular molecule with the same SMILES are termed canonicalization algorithms (Weininger, 1988). Nevertheless, different algorithm implementations may still yield diverse SMILES.

Molecular Attribute Fine-tuning through Reinforcement Learning

In this part, we load the pre-trained transformer network and fine-tune it based on RL. Here, our task is to generate some specific molecules with good attributes. Therefore, we use the generated molecules to measure the properties of the corresponding molecules through Oracle, and use them as rewards to finetune the neural network.

Assume an Agent that must decide on an action $a \in \mathbb{A}(s)$ to take given a particular state $s \in \mathbb{S}$, where $\mathbb{S}$ denotes the set of possible states and $\mathbb{A}(s)$ represents the set of potential actions for that state. The policy $\pi(a \mid s)$ of an Agent associates a state to the likelihood of each action executed within. Reinforcement learning challenges are often depicted as Markov decision processes, indicating that the current state provides all essential information to inform our action choice, and no additional benefit is gained from knowing past states' history. While this is more of an approximation than a fact for most real-life challenges, we can extend this concept to a partially observable Markov decision process where the Agent interacts with a partial environment representation. Let $r(a \mid s)$ be the reward serving as an indicator of the effectiveness of an action taken at a certain state, and the long-term return $G\left(a_t, S_t\right) = \sum_t^T r_t$. represents the cumulative rewards collected from time t to time T. As molecular desirability is only meaningful for a completed SMILES, we will only consider a complete sequence's return.

The main objective of reinforcement learning is to enhance the Agent's policy to increase the expected return $\mathbb{E}[G]$ based on a set of actions taken from some states and the obtained rewards. A task with a definitive endpoint at step T is known as an episodic task (Sutton & Barto, 1999), where T corresponds to the episode's length. SMILES generation is an example of an episodic task, which concludes once the EOS token is sampled.

The states and actions used for Agent training can be produced by the agent itself or through other means. If the agent generates them, the learning is called on-policy, and if generated by other means, it is off-policy learning (Sutton & Barto, 1999).

Reinforcement learning commonly employs two different strategies to determine a policy: value-based RL and policy-based RL (Sutton & Barto, 1999). In value-based RL, the aim is to learn a value function that describes a given state's expected return. Once this function is learned, a policy can be established to maximize a certain action's expected state value. In contrast, policy-based RL aims to learn a policy directly. For the problem we are addressing, we believe policy-based methods are the most suitable for the following reasons:

- Policy-based methods can explicitly learn an optimal stochastic policy (Sutton & Barto, 1999), which aligns with our objective.

- The used method starts with a prior sequence model. The goal is to fine-tune this model based on a specific scoring function. Since the prior model already embodies a policy, fine-tuning might require only minimal changes to the prior model. The short and fast-sampling episodes in this case decrease the gradient estimate's variance impact.

Experiment

Dataset

For any method that necessitates a database, we exclusively use the ZINC 250K dataset [1]. This dataset comprises approximately 250K molecules, selected from the ZINC database due to their pharmaceutical significance, manageable size, and widespread recognition. Both Screening and MolPAL conduct searches within this database. Additionally, generative models like VAEs and LSTMs are pretrained on it. Any fragments essential for JT-VAE (Jin et al., 2018), MIMOSA (Fu et al., 2021), and DST (Fu et al., 2022) are also derived from this very database.

Baseline

To make a comprehensive comparison, eight baseline methods are adopted in performance evaluation.

First, we compare two rule-based baselines, including:

- **REINVENT (Olivecrona et al., 2017).**
  - *Overview:* A method that employs a policy-based reinforcement learning approach to instruct RNNs to produce SMILES strings.
  - *Technical Details:* It formulates the molecular design as a Markov decision process (MDP), where states represent partially generated molecules, and actions correspond to string-based manipulations. The rewards arise from properties of interest in the generated molecules.
  - *Advantage:* Adaptable and can be modified to generate other string representations, such as SELFIES.
  - *Disadvantage:* Relies heavily on the definition and design of rewards.
- **Graph-GA (Jensen, 2019).**
  - *Overview:* A genetic algorithm that manipulates molecular representations using graphs. It integrates crossover operations derived from graph matching and comprises both atom- and fragment-level mutations.
  - *Technical Details:* Unlike string-based genetic algorithms, which primarily involve mutation steps, Graph-GA introduces crossover operations based on graph representations.
  - *Advantage:* Offers a richer set of operations due to its graph-based nature, potentially exploring more diverse chemical spaces.
  - *Disadvantage:* Increased complexity due to the need for graph-based operations.
- **SELFIES-REINVENT.**
  - *Overview:* An extension of REINVENT to generate SELF-referencing Embedded Strings (SELFIES).
  - *Technical Details:* Like REINVENT, it utilizes a policy-based RL approach but specifically for the SELFIES representation, ensuring syntactical validity.
  - *Advantage:* Can produce molecules with fewer syntactical errors due to the nature of SELFIES.
  - *Disadvantage:* Still dependent on the definition of the reward system.
- **GP BO (Tripp et al., 2021).**
  - *Overview:* Incorporates Gaussian process Bayesian optimization, combining surrogate GP models with Graph-GA methods in inner loops.
  - *Technical Details:* While BO traditionally employs non-parametric models, GP BO leverages the GP acquisition function and integrates Graph-GA techniques for sampling.
  - *Advantage:* Balances exploration and exploitation by combining Bayesian optimization with genetic algorithms.
  - *Disadvantage:* The interplay between GP and GA might lead to higher computational costs.

[1] https://www.kaggle.com/datasets/basu369victor/zinc250k

- **STONED (Nigam et al., 2021).**
  - *Overview:* A modified genetic algorithm that manipulates tokens within the SELFIES strings representation.
  - *Technical Details:* Unlike traditional string-based GAs, STONED directly interacts with the tokens in the SELFIES strings.
  - *Advantage:* A more direct approach that can potentially reduce invalid chemical representations.
  - *Disadvantage:* Limited to SELFIES, may not generalize to other representations.
- **SMILES-LSTM HC (Brown et al., 2019).**
  - *Overview:* An iterative learning method leveraging LSTM to comprehend the molecular distribution represented in SMILES strings.
  - *Technical Details:* Uses a variant of the cross-entropy method and integrates generated high-scoring molecules into training data, subsequently fine-tuning the model.
  - *Advantage:* The iterative approach refines the generative process at each step.
  - *Disadvantage:* Convergence might be slow if the initial model is far from optimal.
- **SMILES-GA (Brown et al., 2019).**
  - *Overview:* Genetic algorithm that defines actions based on SMILES context-free grammar.
  - *Technical Details:* Works on SMILES strings and implements genetic mutations and crossovers based on their grammar.
  - *Advantage:* Exploits the inherent structure of SMILES for effective exploration.
  - *Disadvantage:* Limited to the nuances of SMILES grammar, possibly missing out on novel structures.
- **SynNet (Gao et al., 2022).**
  - *Overview:* A synthesis-based genetic algorithm that operates on binary fingerprints and decodes to synthetic pathways.
  - *Technical Details:* Focuses on the synthesis pathway, ensuring the synthesizability of generated molecules.
  - *Advantage:* Prioritizes synthesizability, ensuring generated molecules can be created in the lab.
  - *Disadvantage:* The emphasis on synthesis may limit the diversity of the molecular space explored.
- **DoG-Gen (Bradshaw et al., 2020).**
  - *Overview:* This method is tailored to learn the distribution of synthetic pathways.
  - *Technical Details:* DoG-Gen represents synthetic pathways as Directed Acyclic Graphs (DAGs) and employs an RNN generator to model the distribution. By focusing on synthetic pathways, the method inherently emphasizes the synthesizability of molecules.
  - *Advantages:* Provides a structured approach to learning synthetic pathways.
  - *Disadvantages:* Reliance on RNNs might lead to issues in capturing very long sequences if not designed effectively.
- **DST (Fu et al., 2022).**
  - *Overview:* DST stands for Differentiable Scaffolding Tree, a gradient ascent method designed for molecular optimization.
  - *Technical Details:* DST abstracts molecular graphs into scaffolding trees and makes use of a graph neural network for gradient estimation. This gradient-focused approach enables fine-tuned molecular modifications based on property landscapes in the chemical space.
  - *Advantages:* Offers a more direct way to optimize molecular structures by computing gradients.
  - *Disadvantages:* The abstraction to scaffolding trees may result in loss of information.

Metric

In order to evaluate both optimization capability and sample efficiency, we utilize the area under the curve (AUC) of the top-K average property value in relation to the number of oracle calls. This metric, which we refer to as AUC top-K, serves as our primary measure of performance. Distinct from the straightforward top-K average property, the AUC provides greater reward to methods that achieve high values with a reduced number of oracle calls. In this paper, we set K at 1, 10, and 100. This choice is motivated by the importance of pinpointing a limited set of unique molecular candidates for subsequent phases of development. We cap the number of oracle calls at 10,000, though we anticipate that effective methods should ideally optimize with just hundreds of calls during experimental evaluations. All AUC values reported have been min-max scaled to fit within the range [0, 1].

Evaluation Results

Our result is shown in Table. 1. From the table, we can observe that our method is better than the baseline method on multiple Oracles, which proves the effectiveness of the transformer in our problem.

Overall Molecular Generation Result

The evaluation results depict a thorough comparison between the REINVENT-Transformer (referred to as REINVENT-Trans) and other prominent models across multiple oracles. **Overall Molecular Generation Result**

Performance Overview

REINVENT-Trans demonstrates its strength in molecular generation, consistently achieving top results in several oracles. For instance, the model achieved the highest performance for 'Albuterol_Similarity', 'Mestranol_Similarity', 'QED', 'Scaffold_Hop', and 'Sitagliptin_MPO'. This suggests that the transformer's architecture potentially excels in capturing intricate molecular patterns and relations, and effectively optimizing towards desired properties.

Comparative Insight

1. **Versus REINVENT (SMILES and SELFIES):** REINVENT-Trans has outperformed the REINVENT model (using SMILES) in multiple instances. However, it's worth noting that in some oracles like 'Osimertinib_MPO', REINVENT achieves a marginally better score. It's also evident that SELFIES representation in REINVENT doesn't always improve the performance as compared to its SMILES counterpart. This underscores the importance of the underlying model's architecture and how different representations can influence its performance.

2. **Graph-based Models:** Both 'Graph GA' and 'GP BO' exhibit competitive performance in certain oracles like 'Amlodipine_MPO' and 'Celecoxib_Rediscovery' respectively. However, their performance isn't consistently at the top across all oracles. This implies that while graph-based models can be effective in certain scenarios, they may not always generalize well across diverse tasks.

3. **Genetic Algorithms:** STONED (using SELFIES representation) achieves the highest score in the 'Fexofenadine_MPO' oracle. Genetic algorithms, despite their inherent stochasticity, have potential in some specific optimization tasks.

Sample Efficiency

The primary metric, AUC top-K, emphasizes not just optimization capability but also sample efficiency. High values in this metric would imply fewer oracle calls, thereby maximizing performance with limited data evaluations. REINVENT-Trans, achieving high scores in several oracles, suggests its effectiveness in rapidly optimizing toward desirable molecular properties without exhaustive database searches.

Future Implications

These results showcase the potential of transformers in the domain of molecular generation. Given the success of transformer models in natural language processing tasks, it's no surprise that their capabilities translate effectively into molecular representations as well.

Method Assembly	REINVENT-Trans SMILES	REINVENT SMILES	Graph GA Fragments	REINVENT SELFIES	GP BO Fragments	STONED SELFIES
Albuterol_Similarity	**0.910± 0.008**	0.882± 0.006	0.838± 0.016	0.826± 0.030	0.898± 0.014	0.745± 0.076
Amlodipine_MPO	0.653± 0.029	0.635± 0.035	**0.661± 0.020**	0.607± 0.014	0.583± 0.044	0.608± 0.046
Celecoxib_Rediscovery	0.457± 0.071	0.713± 0.067	0.630± 0.097	0.573± 0.043	**0.723± 0.053**	0.382± 0.041
DRD2	0.931± 0.006	0.945± 0.007	0.964± 0.012	0.943± 0.005	0.923± 0.017	0.913± 0.020
Deco_Hop	0.645± 0.038	0.666± 0.044	0.619± 0.004	0.631± 0.012	0.629± 0.018	0.611± 0.008
Fexofenadine_MPO	0.796± 0.007	0.784± 0.006	0.760± 0.011	0.741± 0.002	0.722± 0.005	**0.797± 0.016**
Isomers_C9H10N2O2PF2Cl	0.809± 0.040	0.642± 0.054	0.719± 0.047	0.733± 0.029	0.469± 0.180	0.805± 0.031
Median 1	0.354± 0.008	**0.356± 0.009**	0.294± 0.021	0.355± 0.011	0.301± 0.014	0.266± 0.016
Median 2	0.263± 0.006	0.276± 0.008	0.273± 0.009	0.255± 0.005	**0.297± 0.009**	0.245± 0.032
Mestranol_Similarity	**0.685± 0.032**	0.618± 0.048	0.579± 0.022	0.620± 0.029	0.627± 0.089	0.609± 0.101
Osimertinib_MPO	0.813± 0.010	**0.837± 0.009**	0.831± 0.005	0.820± 0.003	0.787± 0.006	0.822± 0.012
Perindopril_MPO	0.525± 0.011	0.537± 0.016	0.538± 0.009	0.517± 0.021	0.493± 0.011	0.488± 0.011
QED	**0.942± 0.000**	0.941± 0.000	0.940± 0.000	0.940± 0.000	0.937± 0.000	0.941± 0.000
Ranolazine_MPO	0.761± 0.012	0.742± 0.009	0.728± 0.012	0.748± 0.018	0.735± 0.013	**0.765± 0.029**
Scaffold_Hop	**0.560± 0.013**	0.536± 0.019	0.517± 0.007	0.525± 0.013	0.548± 0.019	0.521± 0.034
Sitagliptin_MPO	**0.563± 0.025**	0.451± 0.003	0.433± 0.075	0.194± 0.121	0.186± 0.055	0.393± 0.083
Thiothixene_Rediscovery	0.556± 0.016	0.534± 0.013	0.479± 0.025	0.495± 0.040	**0.559± 0.027**	0.367± 0.027
Troglitazone_Rediscovery	**0.451± 0.015**	0.441± 0.032	0.390± 0.016	0.348± 0.012	0.410± 0.015	0.320± 0.018
Valsartan_Smarts	**0.165± 0.278**	0.165± 0.358	0.000± 0.000	0.000± 0.000	0.000± 0.000	0.000± 0.000
Zaleplon_MPO	**0.544 ± 0.041**	0.358± 0.062	0.346± 0.032	0.333± 0.026	0.221± 0.072	0.325± 0.027
sum	12.197	12.047	11.526	11.092	11.152	10.598
rank	1	2	3	5	4	6

Method Assembly	LSTM HC SMILES	SMILES GA SMILES	SynNet Synthesis	DoG-Gen Synthesis	DST Fragments
Albuterol_similarity	0.719± 0.018	0.661± 0.066	0.584± 0.039	0.676± 0.013	0.619± 0.020
Amlodipine_MPO	0.593± 0.016	0.549± 0.009	0.565± 0.007	0.536± 0.003	0.516± 0.007
Celecoxib_Rediscovery	0.539± 0.018	0.344± 0.027	0.441± 0.027	0.464± 0.009	0.380± 0.006
DRD2	0.919± 0.015	0.908± 0.019	**0.969± 0.004**	0.948± 0.001	0.820± 0.014
Deco_Hop	**0.826± 0.017**	0.611± 0.006	0.613± 0.009	0.800± 0.007	0.608± 0.008
Fexofenadine_MPO	0.725± 0.003	0.721± 0.015	0.761± 0.015	0.695± 0.003	0.725± 0.005
Isomers_C9H10N2O2PF2Cl	0.342± 0.027	**0.860± 0.065**	0.241± 0.064	0.199± 0.016	0.458± 0.063
Median 1	0.255± 0.010	0.192± 0.012	0.218± 0.008	0.217± 0.001	0.232± 0.009
Median 2	0.248± 0.008	0.198± 0.005	0.235± 0.006	0.212± 0.000	0.185± 0.020
Mestranol_Similarity	0.526± 0.032	0.469± 0.029	0.399± 0.021	0.437± 0.007	0.450± 0.027
Osimertinib_MPO	0.796± 0.002	0.817± 0.011	0.796± 0.003	0.774± 0.002	0.785± 0.004
Perindopril_MPO	0.489± 0.007	0.447± 0.013	**0.557± 0.011**	0.474± 0.002	0.462± 0.008
QED	0.939± 0.000	0.940± 0.000	0.941± 0.000	0.934± 0.000	0.938± 0.000
Ranolazine_MPO	0.714± 0.008	0.699± 0.026	0.741± 0.010	0.711± 0.006	0.632± 0.054
Scaffold_Hop	0.533± 0.012	0.494± 0.011	0.502± 0.012	0.515± 0.005	0.497± 0.004
Sitagliptin_MPO	0.066± 0.019	0.363± 0.057	0.025± 0.014	0.048± 0.008	0.075± 0.032
Thiothixene_Rediscovery	0.438± 0.008	0.315± 0.017	0.401± 0.019	0.375± 0.004	0.366± 0.006
Troglitazone_Rediscovery	0.354± 0.016	0.263± 0.024	0.283± 0.008	0.416± 0.019	0.279± 0.019
Valsartan_Smarts	0.000± 0.000	0.000± 0.000	0.000± 0.000	0.000± 0.000	0.000± 0.000
Zaleplon_MPO	0.206± 0.006	0.334± 0.041	0.341± 0.011	0.123± 0.016	0.176± 0.045
sum	10.227	10.185	9.613	9.554	9.203
rank	7	8	9	10	11

Table 1: Performance comparison between Reinvent-Trans, REINVENT, and other methods over all oracles for AUC Top-10

However, it's essential to consider the versatility of the tasks presented. Some models might specialize in particular tasks but may not be universally applicable. Hence, it's beneficial to have an ensemble or a selection mechanism based on the specific task at hand.

Ablation Study: Long Sequence Molecule Generation Comparison with REINVENT SMILES

The box plot visualizes the distribution of evaluation scores across different molecular lengths for both the reinvent_transformer method and the baseline reinvent method.

Oracle	Model	Avg SA↓	Diversity Top100 ↑
Albuterol Similarity	reinvent	3.177	0.394
	reinvent trans	**3.173**	**0.408**
Amlodipine MPO	reinvent	**3.478**	**0.391**
	reinvent trans	3.888	0.311
Celecoxib Rediscovery	reinvent	3.458	**0.551**
	reinvent trans	**3.245**	0.357
DRD2	reinvent	**2.788**	**0.868**
	reinvent trans	2.914	0.464
Deco Hop	reinvent	3.458	**0.551**
	reinvent trans	**3.240**	0.457
Fexofenadine MPO	reinvent	4.163	0.325
	reinvent trans	**4.113**	**0.411**
GSK3B	reinvent	**3.146**	**0.884**
	reinvent trans	**3.146**	**0.884**
Isomers C7H8N2O2	reinvent	4.273	0.712
	reinvent trans	**2.589**	**0.796**
Isomers C9H10N2O2PF2Cl	reinvent	3.261	0.585
	reinvent trans	**3.245**	**0.686**
Median 1	reinvent	4.571	**0.408**
	reinvent trans	**3.532**	0.371
Median 2	reinvent	**2.772**	**0.411**
	reinvent trans	2.877	0.389
Mestranol Similarity	reinvent	**3.799**	0.267
	reinvent trans	4.394	**0.434**
Osimertinib MPO	reinvent	**3.174**	**0.504**
	reinvent trans	3.799	0.447
Perindopril MPO	reinvent	3.819	**0.479**
	reinvent trans	**3.766**	0.357
QED	reinvent	**1.883**	**0.573**
	reinvent trans	3.422	0.540
Ranolazine MPO	reinvent	3.468	0.421
	reinvent trans	**2.727**	**0.434**
Scaffold Hop	reinvent	**2.857**	**0.555**
	reinvent trans	4.355	0.382
Sitagliptin MPO	reinvent	**2.639**	**0.692**
	reinvent trans	5.279	0.391
Thiothixene Rediscovery	reinvent	**2.899**	0.373
	reinvent trans	3.275	**0.441**
Troglitazone Rediscovery	reinvent	**3.275**	**0.441**
	reinvent trans	4.435	0.204
Valsartan Smarts	reinvent	3.421	0.874
	reinvent trans	3.421	0.874
Zaleplon MPO	reinvent	**1.991**	**0.614**
	reinvent trans	2.465	0.486

Table 2: Avg SA and Diversity Top100

Based on the visual representation, we can derive the following observations:

1. For shorter molecular lengths, both methods exhibit similar distributions of scores.
2. As the molecular length increases, the reinvent_transformer method consistently achieves higher averge scores.
3. For longer molecular sequences, the difference in scores between the two methods becomes more pronounced. This suggests that the reinvent_transformer method is better suited for longer sequences, maintaining high evaluation

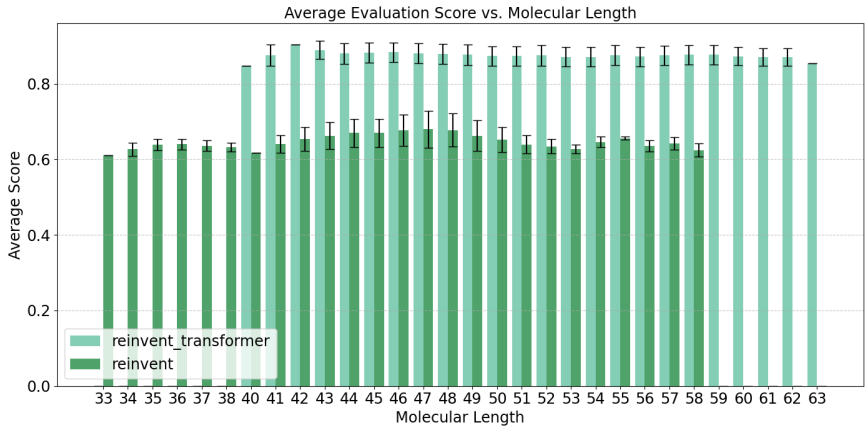

Figure 2: Evaluation score vs molecular length for comparison of Reinvent_Transformer and Reinvent on oracle Mestranol_Similarity

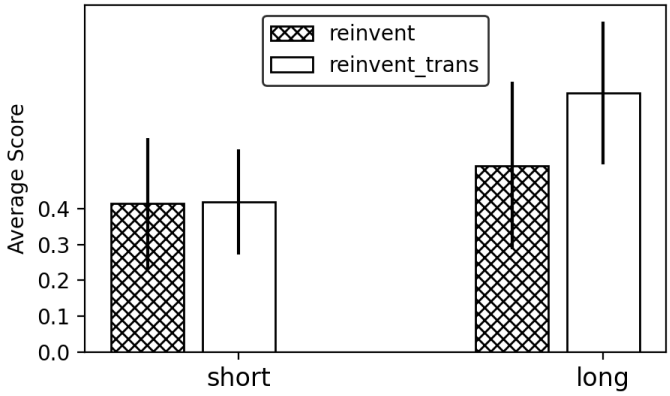

Figure 3: Evaluation score vs short and long sequence for comparison of Reinvent_Transformer and Reinvent on oracle Mestranol_Similarity

scores.
4. The spread (interquartile range) of scores for the reinvent_transformer method remains relatively consistent across molecular lengths, indicating stable performance.

For shorter molecular lengths, both methods exhibit similar distributions of scores. As the molecular length increases, the reinvent_transformer method consistently achieves higher median scores. For longer molecular sequences, the difference in scores between the two methods becomes more pronounced. This suggests that the reinvent_transformer method is better suited for longer sequences, maintaining high evaluation scores. The spread (interquartile range) of scores for the reinvent_transformer method remains relatively consistent across molecular lengths, indicating stable performance.

In conclusion, the reinvent_transformer method outperforms the baseline reinvent method, particularly in the context of longer molecular sequences.

We set a threshold=50 for the length of generated molecular string. If the generated string is longer than the threshold, it will be considered as "long", other it's considered as "short" . From the Figure 3, we can see the our method Reinvent-Transformer has better average score when generating long sequences.

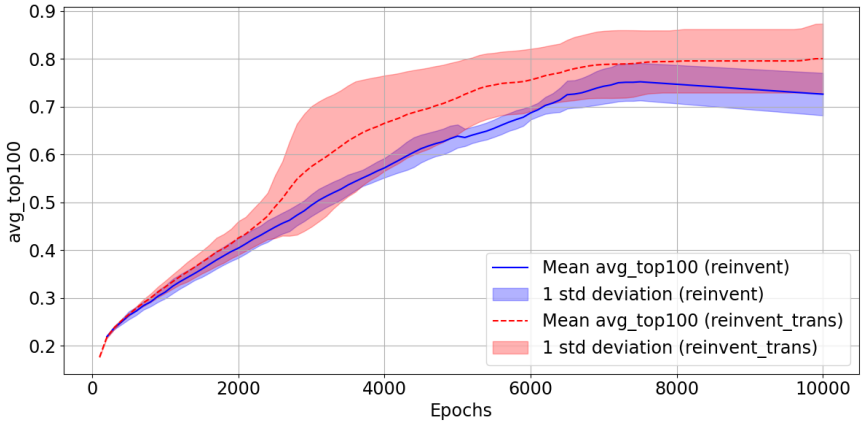

Figure 4: Mean and Standard Deviation of avg_top100 over Epochs for Reinvent and Reinvent-Transformer on oracle Mestranol_Similarity

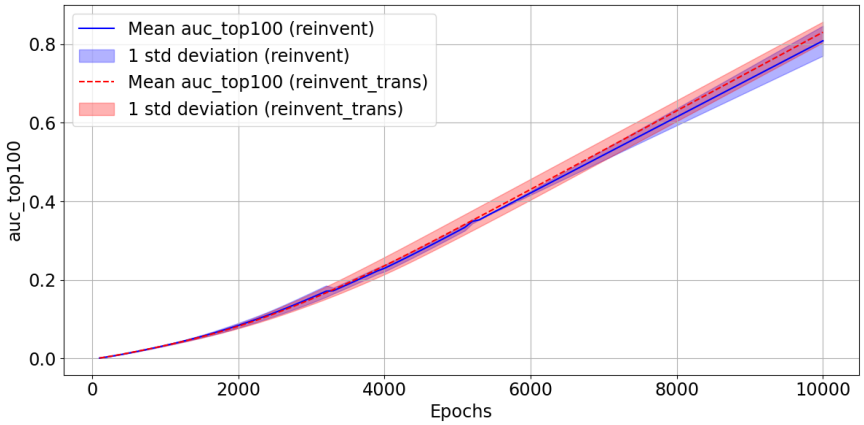

Figure 5: Mean and Standard Deviation of auc_top100 over Epochs for REINVENT and REINVENT-Transformer on oracle Albuterol_Similarity

Case Study: Convergence rate Comparison between Reinvent-Transformer and Reinvent

We plotted the auc_topk curve and number of epoches is the x-axis. From the figure as follows, we can see that our method Reinvent-Transformer converges faster than Reinvent method.

From Fig. 4, the evolution of the average accuracy for the top 100 predictions is evident. Upon examination, across equivalent epochs, the mean accuracy of Reinvent-Transformer consistently surpasses that of Reinvent. This indicates a more expedient convergence rate for the Reinvent-Transformer compared to Reinvent. The avg_top100 curve initially displays a steep incline, eventually plateauing post approximately 6000 epochs. Notably, beginning from the 2500th epoch, the performance differential between Reinvent-Transformer and Reinvent significantly widens.

It is also observed that the Reinvent-Transformer possesses a higher standard deviation relative to Reinvent, suggesting potential variability in its performance. Despite this, the difference between the average top100 accuracy and the standard deviation for Reinvent-Transformer remains superior to the mean accuracy of Reinvent, reaffirming the enhanced efficacy of the Reinvent-Transformer method.

Furthermore, the AUC top100 curve for Albuterol Similarity is illustrated in Fig. 5. In this context, the differential in performance between REINVENT-Transformer and REINVENT is more nuanced. It isn't until the 8000th epoch that a discernible gap emerges. Ultimately, the REINVENT-Transformer exhibits marginally superior performance relative to Reinvent in this scenario.

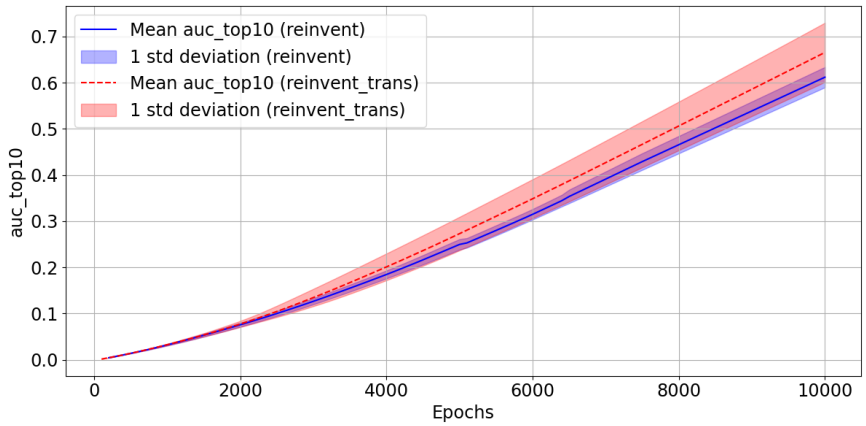

Figure 6: Mean and Standard Deviation of auc_top10 over Epochs for Reinvent and Reinvent-Transformer on oracle Mestranol_Similarity

In Fig. 6, the AUC top10 curve for Mestranol Similarity is presented. Contrasted with the average accuracy curve, this AUC curve demonstrates a milder inclination initially, followed by a pronounced rise. Specifically, for the Reinvent-Transformer, the mean AUC top10 consistently surpasses that of Reinvent. Although the disparity is subtle during the initial epochs, it becomes more pronounced post the 5000th epoch and remains so thereafter.

Conclusion

Navigating the vast chemical space in molecular design remains a challenge. The introduction of the REINVENT-Transformer marks a significant advancement, harnessing the Transformer architecture's strengths such as parallelization and long-term dependency handling. Our experimental findings reinforce the REINVENT-Transformer's superior performance across multiple oracles, especially in tasks requiring longer sequence data. By integrating oracle feedback reinforcement learning, our approach achieves heightened precision, favorably impacting drug discovery efforts. In essence, the REINVENT-Transformer not only sets a benchmark in molecular de novo design but also illuminates the path for future research, highlighting the promise of Transformer-based architectures in drug discovery.

References

Mathieu Bouville. Crime and punishment in scientific research, 2008.

John Bradshaw, Brooks Paige, Matt J Kusner, Marwin Segler, and José Miguel Hernández-Lobato. Barking up the right tree: an approach to search over molecule synthesis dags. *Advances in Neural Information Processing Systems*, 33:6852–6866, 2020.

Nathan Brown, Marco Fiscato, Marwin HS Segler, and Alain C Vaucher. GuacaMol: benchmarking models for de novo molecular design. *Journal of chemical information and modeling*, 59(3):1096–1108, 2019.

William J. Clancey. *Transfer of Rule-Based Expertise through a Tutorial Dialogue*. Ph.D. diss., Dept. of Computer Science, Stanford Univ., Stanford, Calif., 1979.

William J. Clancey. Communication, Simulation, and Intelligent Agents: Implications of Personal Intelligent Machines for Medical Education. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence (IJCAI-83)*, pp. 556–560, Menlo Park, Calif, 1983. IJCAI Organization.

William J. Clancey. Classification Problem Solving. In *Proceedings of the Fourth National Conference on Artificial Intelligence*, pp. 45–54, Menlo Park, Calif., 1984. AAAI Press.

William J. Clancey. The Engineering of Qualitative Models. Forthcoming, 2021.

Robert Engelmore and Anthony Morgan (eds.). *Blackboard Systems*. Addison-Wesley, Reading, Mass., 1986.

Tianfan Fu, Cao Xiao, Xinhao Li, Lucas M Glass, and Jimeng Sun. MIMOSA: Multi-constraint molecule sampling for molecule optimization. *AAAI*, 2021.

Tianfan Fu, Wenhao Gao, Cao Xiao, Jacob Yasonik, Connor W Coley, and Jimeng Sun. Differentiable scaffolding tree for molecular optimization. *International Conference on Learning Representations*, 2022.

Wenhao Gao, Rocío Mercado, and Connor W Coley. Amortized tree generation for bottom-up synthesis planning and synthesizable molecular design. *International Conference on Learning Representations*, 2022.

Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 2018.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

Diane Warner Hasling, William J. Clancey, Glenn R. Rennels, and Thomas Test. Strategic Explanations in Consultation—Duplicate. *The International Journal of Man-Machine Studies*, 20(1):3–19, 1983.

Diane Warner Hasling, William J. Clancey, and Glenn Rennels. Strategic explanations for a diagnostic consultation system. *International Journal of Man-Machine Studies*, 20(1):3–19, 1984. ISSN 0020-7373. doi: https://doi.org/10.1016/S0020-7373(84)80003-6. URL `https://www.sciencedirect.com/science/article/pii/S0020737384800036`.

Jan H Jensen. A graph-based genetic algorithm and generative model/monte carlo tree search for the exploration of chemical space. *Chemical science*, 10(12):3567–3572, 2019.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. *ICML*, 2018.

Ksenia Korovina, Sailun Xu, Kirthevasan Kandasamy, Willie Neiswanger, Barnabas Poczos, Jeff Schneider, and Eric Xing. ChemBO: Bayesian optimization of small organic molecules with synthesizable recommendations. In *International Conference on Artificial Intelligence and Statistics*, pp. 3393–3403. PMLR, 2020.

Mario Krenn, Florian Häse, AkshatKumar Nigam, Pascal Friederich, and Alan Aspuru-Guzik. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, 1(4):045024, 2020.

Henry Moss, David Leslie, Daniel Beck, Javier Gonzalez, and Paul Rayson. BOSS: Bayesian optimization over string spaces. *Advances in neural information processing systems*, 33:15476–15486, 2020.

NASA. Pluto: The 'other' red planet. https://www.nasa.gov/nh/pluto-the-other-red-planet, 2015. Accessed: 2018-12-06.

AkshatKumar Nigam, Robert Pollice, Mario Krenn, Gabriel dos Passos Gomes, and Alan Aspuru-Guzik. Beyond generative models: superfast traversal, optimization, novelty, exploration and discovery (STONED) algorithm for molecules using SELFIES. *Chemical science*, 12(20):7079–7090, 2021.

Marcus Olivecrona, Thomas Blaschke, Ola Engkvist, and Hongming Chen. Molecular de-novo design through deep reinforcement learning. *Journal of cheminformatics*, 9(1):1–14, 2017.

James Rice. Poligon: A System for Parallel Problem Solving. Technical Report KSL-86-19, Dept. of Computer Science, Stanford Univ., 1986.

Arthur L. Robinson. New ways to make microcircuits smaller. *Science*, 208(4447):1019–1022, 1980a. ISSN 0036-8075. doi: 10.1126/science.208.4447.1019. URL https://science.sciencemag.org/content/208/4447/1019.

Arthur L. Robinson. New Ways to Make Microcircuits Smaller—Duplicate Entry. *Science*, 208:1019–1026, 1980b.

Cynthia Shen, Mario Krenn, Sagi Eppel, and Alan Aspuru-Guzik. Deep molecular dreaming: Inverse machine learning for de-novo molecular design and interpretability with surjective representations. *Machine Learning: Science and Technology*, 2021.

Richard S Sutton and Andrew G Barto. Reinforcement learning: An introduction. *Robotica*, 17(2):229–235, 1999.

Austin Tripp, Gregor NC Simm, and José Miguel Hernández-Lobato. A fresh look at de novo molecular design benchmarks. In *NeurIPS 2021 AI for Science Workshop*, 2021.

David Weininger. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.

David Weininger. Smiles. Accessed 7 April 2017, 2017. http://www.daylight.com/dayhtml/doc/theory/theory.smiles.html.

Zhenpeng Zhou, Steven Kearnes, Li Li, Richard N Zare, and Patrick Riley. Optimization of molecules via deep reinforcement learning. *Scientific reports*, 9(1):1–10, 2019.