Projecting Molecules into Synthesizable Chemical Spaces

Shitong Luo [* 1] Wenhao Gao [* 2] Zuofan Wu [* 1] Jian Peng [1] Connor W. Coley [2] Jianzhu Ma [3]

Abstract

Discovering new drug molecules is a pivotal yet challenging process due to the near-infinitely large chemical space and notorious demands on time and resources. Numerous generative models have recently been introduced to accelerate the drug discovery process, but their progression to experimental validation remains limited, largely due to a lack of consideration for synthetic accessibility in practical settings. In this work, we introduce a novel framework that is capable of generating new chemical structures while ensuring synthetic accessibility. Specifically, we introduce a postfix notation of synthetic pathways to represent molecules in chemical space. Then, we design a transformer-based model to translate molecular graphs into postfix notations of synthesis. We highlight the model's ability to: (a) perform bottom-up synthesis planning more accurately, (b) generate structurally similar, synthesizable analogs for unsynthesizable molecules proposed by generative models with their properties preserved, and (c) explore the local synthesizable chemical space around hit molecules.

1. Introduction

Designing novel drug molecules, a process that aims to identify molecular structures with desired properties, is known to be notoriously complex, time- and resource-consuming (Wouters et al., 2020). Methods that could accelerate the discovery process are thus of significant interest in medicinal science and the pharmaceutical industry. Historically, two principal computational methodologies have been employed in drug design: virtual screening (Shoichet, 2004) and *de novo* design (Hartenfeller & Schneider, 2011). Virtual screening involves predicting properties for a predefined set of molecules, known as virtual libraries, and identifying the most promising candidates for subsequent validation. These libraries are usually commercial catalogs (make-on-demand compounds) from chemical vendors (Lyu et al., 2019) and generally consist of molecules with assured synthesizability (Enamine, 2023). However, screening large numbers of molecules is a time-intensive process. Moreover, even libraries on the scale of billions present only a tiny fraction of the theoretical chemical space (Figure 1a), which is nearly infinite in the number of potential synthesizable small molecules (Bohacek et al., 1996). This means that many potentially valuable molecules might not be included in these libraries. Additionally, as the size of the library increases, so does the cost of screening, which in turn limits the effectiveness and efficiency of virtual screening in the drug discovery process.

De novo molecular design methods, unlike virtual screening, dynamically create molecules tailored to specific objectives like protein binding and bioactivity. Early methods of this kind typically adopt combinatorial optimization (Venkatasubramanian et al., 1995; Jensen, 2019; Brown et al., 2019), while recently, neural network enhanced optimization (You et al., 2018; Fu et al., 2021) and generative modeling (Gómez-Bombarelli et al., 2018; Jin et al., 2018) have become popular. These methods aren't constrained by the limitations of virtual libraries, enabling the efficient exploration of novel chemical entities (see Figure 1c). However, a notable defect with most existing generative models is their tendency to propose synthetically infeasible molecules (Figure 1c) (Gao & Coley, 2020), a concern particularly pronounced in structure-based drug design, an area of increasing interest (Bohacek et al., 1996; Luo et al., 2021; Fu et al., 2022). This issue stems from the inherent ignorance of synthetic accessibility in formulations of most existing generative algorithms, in which molecules are constructed using SMILES strings or atoms and motifs as fundamental units (Gao et al., 2022). This often leads to chemical structures that fall outside synthesizable chemical spaces, posing practical challenges in the experimental validation as well as production of such designs.

Incorporating synthesizability as an additional optimization criterion is a common strategy, yet it introduces complex challenges. Synthesizability is difficult to define precisely due to its sensitivity to subtle structural differences, reaction selectivity, and building block availability inherent in chem-

ical reactions (Gao & Coley, 2020). While notable efforts have been made to estimate synthetic accessibility (Ertl & Schuffenhauer, 2009; Thakkar et al., 2021; Voršilák et al., 2020), reliable quantification of synthetic accessibility is still a far goal. A more effective strategy involves constraining the design process to only consider molecules that are synthesizable, ensuring that algorithms focus exclusively on viable structures. Based on this idea, several models have been introduced (Bradshaw et al., 2019; 2020; Gao et al., 2021; Swanson et al., 2023), though these models have notable limitations, such as poor coverage of synthesizable chemical space or inefficiency in optimizing the properties, as further discussed in Section 2.

To address the recognized challenges, we introduce a framework that learns to "project" molecules into synthesizable chemical spaces (Figure 1d). Given molecules without synthesizability guarantee, our model is able to "fix" them by identifying synthesizable analogs that are structurally similar, thus preserving the structural factors that contributed to the desired properties (Janela & Bajorath, 2022). We represent molecules by a scalable linear representation for synthetic pathways based on postfix notations. By generating synthetic pathways in the form of postfix notations rather than molecular graphs or strings, we can ensure the designed molecules are derivable from purchasable chemical building blocks and expert-defined chemical reaction rules, thereby guaranteeing the synthesizability of resulting molecules. We adopt the transformer architecture (Vaswani et al., 2017), with an encoder for molecular graphs and a decoder to generate postfix notations of synthesis.

We validate the effectiveness of the model by showcasing its ability to generate synthesizable analogs that are structurally and functionally similar to input unsynthesizable molecules. Furthermore, the model demonstrates its proficiency in navigating the local synthesizable chemical space, proving invaluable for the expansion and exploration of potential hit molecules. The main contributions of our work can be summarized as follows:

- We propose a novel linear representation for synthesis pathways based on postfix notations. This also provides a uniform and generalizable representation for all experimental procedures.

- We present a novel and robust algorithm tailored for synthesizable molecular design. Specifically, our model can project unsynthesizable molecules from existing generative models to synthesizable chemical space to obtain a structural analog with desired properties such as binding and bioactivity preserved, as well as explore local chemical space for hit expansion.

- We demonstrate strong empirical performance in evaluations of bottom-up synthesis planning and analog design, outperforming prior methods.
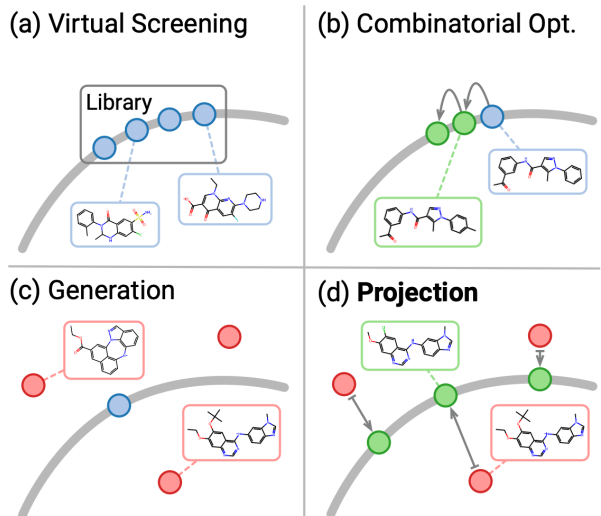

Figure 1. Illustration of **(a)** virtual screening algorithms, **(b)** combinatorial optimization algorithms, **(c)** *de novo* generative models, and **(d)** projection to synthesizable chemical space. Each dot represents a chemical structure, and the gray curve represents a synthesizable chemical space.

2. Related Work

We divide the synthesizable molecular design methods into combinatorial optimization algorithms, reinforcement learning, and autoencoders. We acknowledge and emphasize that the boundaries between these categories are not clear-cut and many intersections exist. This classification is primarily for the sake of clarity in our discussion.

Combinatorial optimization algorithms use a set of chemical building blocks and reaction templates to make the chemical space combinatorial, and then apply combinatorial optimization algorithms, such as genetic algorithms (Gao et al., 2021), Monte Carlo tree search (Swanson et al., 2023), and Bayesian optimization (Korovina et al., 2020), to search for desired molecules (Figure 1b). Early representative work before the advent of molecular deep learning includes SYNOPSIS (Vinkers et al., 2003) and DOGS (Hartenfeller et al., 2012), which perform virtual reactions to generate candidate molecules and select candidates according to scoring functions. In recent years, advances in deep learning for chemical reactions (Coley et al., 2019a; Schwaller et al., 2019) have inspired methods such as MoleculeChef (Bradshaw et al., 2019), ChemBO (Korovina et al., 2020), and DoGs (Bradshaw et al., 2020) that use neural networks to predict reaction outcomes in a "template-free" manner.

Reinforcement learning approaches, exemplified by PGFS (Gottipati et al., 2020) and REACTOR (Horwood & Noutahi, 2020), formulate the forward synthesis process as Markov decision processes and train agents to explore the chemical space under the guidance of reward functions that

quantify desired properties. The agent is provided with a set of reactants and reaction templates and learns to synthesize desirable molecules with them. As the agent is aware of the objective, its search process is directed and expected to be more efficient.

Autoencoder-based methods, represented by MoleculeChef (Bradshaw et al., 2019) and DoGs (Bradshaw et al., 2020), leverage the autoencoder architecture to model synthetic pathways in a generative manner. MoleculeChef is one of the earliest neural models of this kind. Instead of synthetic paths, it tries to encode and decode a bag of starting reactants from purchasable building blocks. Following an enumeration of the one-step synthetic pathways, the "best" molecule from the products is chosen as the output. DoGs represents synthetic pathways as directed acyclic graphs and serializes the graph construction process as a sequence of actions that includes node addition, reactant selection, and node connection. DoGs use recurrent neural networks to generate action sequences from latent codes, enabling *de novo* molecular generation by drawing samples from the latent space. SynNet (Gao et al., 2021) can also be seen as this kind, in which the molecular fingerprinting algorithm is the encoder, and neural networks are used as a decoder to generate synthetic paths.

Limitations: While the synthesizable molecular design methods have advanced significantly, they are still in the proof-of-concept stage and far from mature. The efficiency of combinatorial optimization often falls short, as these algorithms edit molecules locally, thus being inefficient to explore the vast chemical space. Reinforcement learning techniques depend heavily on querying oracle functions, which can hinder their practical application. MoleculeChef (Bradshaw et al., 2019) primarily explores one-step pathways due to the computational impracticality of enumerating an exponentially growing number of candidates. Models like PGFS (Gottipati et al., 2020) and REACTOR (Horwood & Noutahi, 2020) are limited to generating only linear synthetic pathways. In addition, the capacity of these models to generalize and construct complex synthetic paths is yet to be convincingly demonstrated. Models such as DoGs (Bradshaw et al., 2020) and SynNet (Gao et al., 2021), theoretically capable of encompassing a complete chemical space, have not been empirically validated for their ability to adapt to novel, unseen chemicals. Additionally, these models are generally less adept at generating convergent pathways or more intricate molecular structures, indicating a need for further refinement in their design and functionality.

3. Method

3.1. Preliminaries

Chemical space A chemical space $\mathcal{C}$ is a set of molecules generated by a finite set of *building block* molecules $\mathcal{B} = \{B_1, B_2, \ldots B_N\} \subseteq \mathcal{C}$ and a set of *reaction* rules $\mathcal{R} = \{R_1, R_2, \ldots R_M\}$. Formally, a reaction rule maps reactants to a reaction product. For example, a reaction rule R with two reactants can be denoted by:

$$\begin{aligned} R : &\mathcal{X}_1 \times \mathcal{X}_2 \to \mathcal{C} \\ &(X_1, X_2) \mapsto Y, \end{aligned} \tag{1}$$

where $\mathcal{X}_1$ and $\mathcal{X}_2$ are sets of molecules to which reaction R can be applied, and Y is the *main* reaction product. The chemical space is generated by starting from the building blocks and iteratively applying the reactions to every possible combination of molecules. There are some cases where a reaction has multiple possible *main* products. For ease of discussion, we assume that reactions produce only one main product in this section, though the actual implementation of our model supports multiple products.

Every molecule in the chemical space is a product of a synthetic process, which involves applying reaction rules to building blocks and intermediate molecules to build a complex molecule. For example, $R_2(R_1(B_1, B_2), B_3)$ denotes a process in which we first apply reaction R_1 to building blocks B_1 and B_2, and then apply reaction R_2 to building block B_3 and the product of $R_1(B_1, B_2)$.

Postfix notation of synthesis We formalize the synthetic process as a sequence of function call instructions with a memory stack to store arguments and intermediate results. The instruction sequence contains two types of operations: pushing a building block molecule onto the stack and applying a reaction. The reaction operation first pops the required number of molecules at the top of the stack, then computes the reaction product which will be pushed back onto the stack. The two operations can be denoted by the building block and the reaction associated with them, leading to the postfix notation of the synthetic process.

Following the above example, the instruction sequence for $R_2(R_1(B_1, B_2), B_3)$ is: (1) push B_1; (2) push B_2; (3) apply R_1; (4) push B_3; (5) apply R_2, which can be simplified as the postfix notation: $[B_1, B_2, R_1, B_3, R_2]$ (Figure 2).

The synthetic process can be parsed to an abstract syntax tree (AST), which is equivalent to the synthetic tree used in previous work and the postfix notation is a post-order traversal sequence of the AST. Although it has connections with previous methods, the proposed postfix notation has multiple advantages. First, the postfix notation is capable of representing branched synthesis pathways, where reactions can be applied to more than one intermediate deposited in
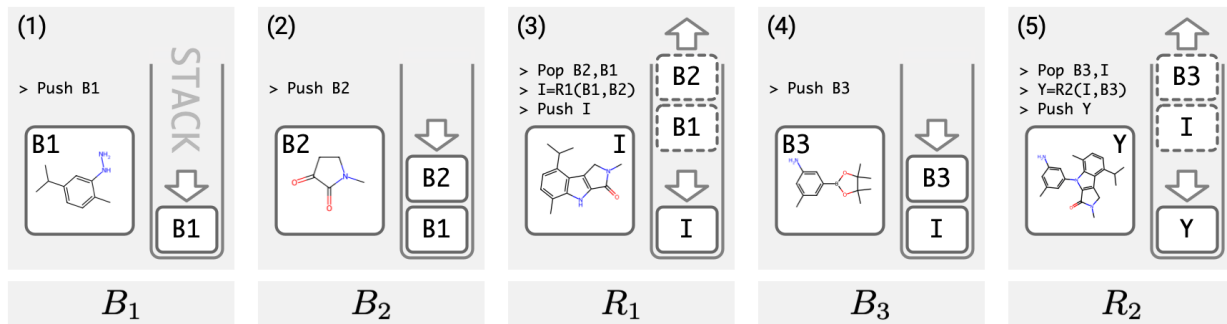
Figure 2. **Top:** Simulation of the synthetic plan $R_2(R_1(B_1, B_2), B_3)$ with a memory stack to store intermediate molecules. In the first two steps, building blocks B_1 and B_2 are pushed onto the stack. In the third step, B_1 and B_2 are popped from the stack and used as input for reaction R_1, leading the intermediate reaction product I, which is then added back onto the stack. Subsequently, in the fourth step, building block B_3 is pushed onto the stack. Finally, reaction R_2 is applied to I and B_3, leading to the final product Y. **Bottom:** The postfix notation of the synthetic pathway is $[B_1, B_2, R_1, B_3, R_2]$. Each symbol in the notation corresponds to one step in the synthesis.

the stack. In contrast, previous methods (Gottipati et al., 2020; Horwood & Noutahi, 2020) are limited to linear synthetic pathways, where intermediates can only react with building blocks. In addition, the postfix notation is a simple representation comprising only building blocks and reactions, whereas previous synthetic tree-based representations require extra symbols, such as node addition and edge addition, to construct a synthetic plan. Lastly, the postfix notation allows reactions with multiple (more than two) molecule inputs, while the previous representation (Gao et al., 2021) is incompatible with such complex reaction rules. Therefore, the postfix notation can be immediately applied to chemical spaces that include reactions with reagent specification or with multiple reactants, showing the potential for more complex settings in future work.

Problem definition We formulate the task of this work, projecting molecules into the chemical space, as generating a postfix notation of synthesis from the input molecular graph, so that the postfix notation denotes a molecule in the chemical space that is identical or structurally similar to the input molecule.

3.2. Model

The model consists of two components: a graph transformer encoder network that encodes the molecular graph and a transformer decoder network that autoregressively generates postfix notations.

Molecular graph encoder A molecule is represented by a graph with a set of atoms and bonds. The graph encoder network first converts each atom into initial embeddings according to the atomic number. Then a stack of graph transformer layers (Ying et al., 2021), which take edges into account by adding a bond type-dependent learnable bias term to the query-key product matrix, performs multi-head self-attention on the atoms. Finally, the output per-atom

embeddings are used for decoding.

Postfix notation decoder A postfix notation of synthesis is a sequence that contains four types of tokens: building block tokens, reaction tokens, a start token [START], and an end token [END]. A building block token is attached with the fingerprint of the building block molecule, denoted by [BB, $\boldsymbol{p}$], where $\boldsymbol{p} \in \{0, 1\}^{256}$ is the Morgan fingerprint of length 256 and radius 2 (Morgan, 1965). Building block tokens are converted into embeddings by an MLP. A reaction token is the index r of the reaction denoted by [RXN, r]. The reaction tokens and the start token are converted into embeddings by looking up from the embedding table. Sinusoidal positional encodings (Vaswani et al., 2017) that indicate the order of tokens are added to the embeddings. In summary, the initial embedding of the i-th token T_i is defined as:

$$\boldsymbol{h}_i^{(0)} = \text{PE}(i) + \begin{cases} \boldsymbol{e}_{\text{start}}, & \text{type}(T_i) = \text{START} \\ \text{MLP}_{\text{fp}}(\boldsymbol{p}_i), & \text{type}(T_i) = \text{BB} \\ \text{Embed}_{\text{rxn}}(r_i), & \text{type}(T_i) = \text{RXN} \end{cases}.$$
$$(2)$$

Next, a stack of standard transformer layers (Vaswani et al., 2017) takes the sequence embeddings along with the molecular graph embeddings as input, and then predicts the next token using the embeddings from the last layer $\boldsymbol{h}_i$ for each position of the sequence. The prediction network first predicts the type of the next token:

$$\text{type}(\hat{T}_{i+1}) \sim \text{softmax}(\text{MLP}_{\text{type}}(\boldsymbol{h}_i)), \qquad (3)$$

where possible token types include BB, RXN, and END. If the next token is predicted to be a reaction token BB, the fingerprint will be predicted and the building block will be
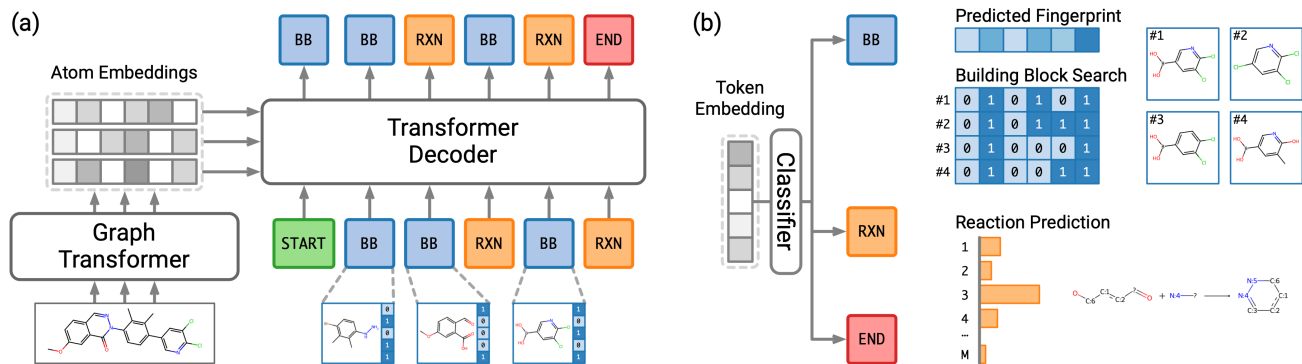
Figure 3. **(a)** Overall architecture of the network. A graph transformer encodes the input molecular graph into atom embeddings. A transformer decoder autoregressively decodes the postfix notation, with atom embeddings as the context. **(b)** Architecture of the prediction network. First, a classifier predicts the token type from the token embedding. If the predicted token type is [BB] (building block), the token embedding is used to predict the building block fingerprint to retrieve building blocks. If the token type is [RXN] (reaction), the token embedding is used to predict the type of reaction. Otherwise, [END] marks the end of the postfix notation.

retrieved via nearest-neighbor search:

$$\hat{\boldsymbol{p}}_{i+1} = \text{sigmoid}(\text{MLP}_{\text{fp-pred}}(\boldsymbol{h}_i)), \tag{4}$$

$$\hat{B}_{i+1} = \text{Nearest-Neighbor}(\hat{\boldsymbol{p}}_{i+1}, \{B_1, \ldots B_N\}), \tag{5}$$

where $\{B_1, \ldots B_N\}$ is the building block set of the chemical space. The fingerprint of the retrieved $\hat{B}_{i+1}$ is used to replace the predicted fingerprint as input to the model before predicting the next token. The nearest-neighbor search could produce multiple building block candidates, enabling the generation of diverse pathways. If the next token is predicted to be a reaction token, the reaction type will be predicted by:

$$\hat{r}_{i+1} \sim \text{softmax}(\text{MLP}_{\text{rxn}}(\boldsymbol{h}_i)). \tag{6}$$

Similarly, the sampled reaction template $\hat{r}_{i+1}$ is used as input to the decoder to predict the next token.

3.3. Training

We sample synthetic pathways from the chemical space and convert product molecules into graphs to create (postfix notation, graph) pairs for training. The synthesis sampling algorithm iteratively selects building blocks and applicable reactions at random to generate valid operation sequences as described in Algorithm 1 in Appendix. Synthetic pathways are generated on-the-fly at each training step.

We use the standard technique to train transformer decoders in parallel (Vaswani et al., 2017) — feeding the complete sequence into the decoder, offseting the output embeddings by one position, and applying causal masking.

The training loss consists of three terms corresponding to the three prediction heads. The first term is a cross entropy

loss for token type:

$$L_{\text{type}} = \frac{1}{\ell} \sum_{i=0}^{\ell-1} \text{CE}\left(\text{type}(\hat{T}_{i+1}), \text{type}(T_{i+1})\right), \tag{7}$$

where ℓ is the length of the sequence. The second term is a loss function for building block fingerprints:

$$L_{\text{bb}} = \frac{1}{n} \sum_{i=0}^{\ell-1} \mathbb{I}\left(\text{type}(T_{i+1}) = \text{BB}\right) \cdot \\ \sum_{j=0}^{255} \text{BCE}\left(\hat{\boldsymbol{p}}_{i+1}[j], \boldsymbol{p}_{i+1}[j]\right), \tag{8}$$

where $\mathbb{I}$ is the indicator function, BCE is the binary cross entropy loss function, and $n = \sum_{i=0}^{\ell-1} \mathbb{I}\left(\text{type}(T_{i+1}) = \text{BB}\right)$ is the number of BB tokens in the sequence. The last term is a cross entropy loss for reaction type:

$$L_{\text{rxn}} = \frac{1}{m} \sum_{i=0}^{\ell-1} \mathbb{I}\left(\text{type}(T_{i+1}) = \text{RXN}\right) \cdot \text{CE}\left(\hat{r}_{i+1}, r_{i+1}\right). \tag{9}$$

The final loss function is the sum of the three terms: $L = L_{\text{type}} + L_{\text{bb}} + L_{\text{rxn}}$.

3.4. Inference

At inference time, we create an empty stack and generate tokens one by one. When the generated token is a building block token, we look up the building block molecule according to the predicted fingerprint and push the building block onto the stack. When the generated token is a reaction token, we first pop molecules from the stack according to the number of required reactants, then use RDKit to predict the product according to the reaction template, and finally

Table 1. The proposed method achieves higher success rate, reconstruction rate, and similarity scores in comparison to the baseline model, indicating that our model can find synthetic pathways for synthesizable molecules more accurately.

Dataset	Method	Success%	Recons.%	Sim.(Morgan)	Sim.(Scaffold)	Sim.(Gobbi)
Test Set	SynNet	84.1%	10.7%	0.4575	0.5109	0.3465
	Proposed	**97.5%**	**28.4%**	**0.7167**	**0.7791**	**0.7273**
ChEMBL	SynNet	85.0%	5.4%	0.4270	0.4174	0.2678
	Proposed	**98.8%**	**13.3%**	**0.5978**	**0.5869**	**0.5570**

push the product onto the stack. If the number of molecules on the stack is less than the required number or if the reaction is not applicable to the molecules, the inference process will be aborted. Finally, an end token `[END]` terminates the process and the molecule at the top of the stack will be taken as the final product. The product molecule is considered to be a *projection* of the input molecular graph in the chemical space. We select the final product molecules according to properties of interest, such as similarity and docking scores.

3.5. Experiment Setup

Reaction templates We use the SynNet reaction template set (Gao et al., 2021) curated by the authors from two publicly available template sets (Hartenfeller et al., 2012; Button et al., 2019). The template set contains 91 reaction templates encoded in SMARTS string format (Daylight Chemical Information Systems, Inc., 2023), of which 13 are uni-molecular reactions and 78 are bi-molecular reactions.

Building blocks We use the building blocks in the Enamine US Stock catalog retrieved in October 2023 (Enamine, 2023). For building blocks that include more than one molecule (*e.g.*, salts, hydrates), we keep the largest molecule and drop the remaining ones. Building blocks that fail the RDKit sanitization check or do not match any reaction template are removed. Duplicate building blocks are also removed. The pre-processing procedure leads to the final building block set containing 211,220 molecules.

Similarity scores To evaluate the similarity between the input molecule and the output molecule, we use the Tanimoto similarity score on three different fingerprints: (1) Morgan fingerprint of length 4096 and radius 2 (Morgan, 1965), (2) Morgan fingerprint of Murcko scaffold, and (3) Gobbi pharmacophore fingerprint (Gobbi & Poppinger, 1998). The three similarity scores indicate chemical similarities in three different aspects: overall structure, scaffold structure, and pharmacophore property, respectively, and are all normalized to $[0, 1]$.

Code and data The code and data of this project are available at `https://github.com/luost26/ChemProjector`.

4. Results

In this section, we first examine the model's capability of finding synthesis pathways for molecules that are known to be synthesizable, and explore the generalization ability of the models (Section 4.1). Then, we study the application of the model in three drug design settings: structure-based drug design (Section 4.2), goal-directed generation (Section 4.3), and hit expansion (Section 4.4).

4.1. Generalization in Bottom-Up Synthesis Planning

We first validate our model's ability in bottom-up synthesis planning, particularly in proposing synthetic pathways for novel molecular structures. To assess this capability with unseen molecules, we employed the following testing approach. First, we clustered building blocks into 128 groups using K-means algorithm based on their fingerprints, reserving one cluster exclusively for testing while using the remaining 127 for training. This approach generated a test set of 1,000 molecules from the reserved building block to evaluate our model's generalization. During testing, we allow the model to access the reserved building blocks. In addition, we include another challenging benchmark: 1,000 molecules from the ChEMBL database (Release 33)(Gaulton et al., 2012), reported to be predominantly "unreachable" in previous work (Gao et al., 2021). This not only tested our model against complex, unseen molecular structures but also set a strenuous standard for assessing its generalizability.

We evaluate the model using the following metrics: (1) success rate: the percentage of valid postfix notations; (2) reconstruction rate: the percentage of proposed synthetic pathways that lead to products identical to input molecules; (3) average similarities. The similarity score is set to 0 for failed cases. We compare our model with the baseline model SynNet (Gao et al., 2021).

Our model produces valid synthetic pathways for most test cases and thus achieves a high success rate (Table 1). Most notably, 29.3% of the test molecules and 13.4% of the ChEMBL molecules have been successfully reconstructed. Furthermore, the similarity scores of the proposed model are significantly higher than the baseline model. The result demonstrates that our model exhibits generalizability to
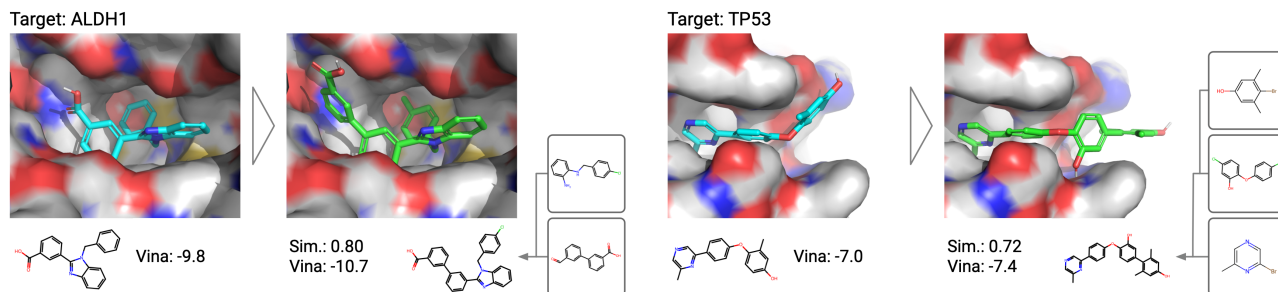
Figure 4. Examples of projecting molecules generated by Pocket2Mol to synthesizable analogs. The projection improves the Vina score and proposes synthesis pathways. The docking poses remain similar.

Table 2. Similarity scores between molecules generated by Pocket2Mol and their analogs.

	Sim. (Morgan)	Sim. (Scaffold)	Sim. (Gobbi)
ADRB2	0.5804	0.6362	0.4409
ALDH1	0.3875	0.3399	0.3034
ESR1_ago	0.4066	0.3520	0.2990
ESR1_ant	0.4965	0.4919	0.4122
FEN1	0.4180	0.4090	0.3369
GBA	0.3140	0.2572	0.2460
IDH1	0.3830	0.3423	0.3178
KAT2A	0.5028	0.4831	0.4378
MAPK1	0.4274	0.3942	0.3886
MTORC1	0.4565	0.4193	0.3820
OPRK1	0.5125	0.5428	0.4312
PKM2	0.4558	0.4315	0.3818
PPARG	0.4992	0.4884	0.4416
TP53	0.5566	0.5386	0.4617
VDR	0.4355	0.3857	0.3615

Table 3. The analogs slightly improve Vina scores, suggesting the binding is likely to be preserved.

Target	Vina (kcal/mol)			
	Ref.	Gen.	Analog ($\downarrow$)	Δ ($\downarrow$)
ADRB2	-8.24	-8.70	-9.09	-0.38
ALDH1	-8.49	-9.10	-9.00	0.10
ESR1_ago	-8.28	-9.46	-9.36	0.09
ESR1_ant	-7.62	-9.77	-9.77	0.00
FEN1	-6.32	-6.54	-6.60	-0.06
GBA	-8.14	-6.96	-7.69	-0.73
IDH1	-9.19	-9.50	-9.56	-0.07
KAT2A	-7.18	-8.07	-8.18	-0.11
MAPK1	-8.63	-8.61	-8.98	-0.37
MTORC1	-9.64	-10.17	-10.20	-0.04
OPRK1	-8.83	-8.36	-8.77	-0.40
PKM2	-9.23	-9.45	-10.09	-0.65
PPARG	-7.16	-8.47	-8.53	-0.06
TP53	-6.13	-7.11	-7.47	-0.36
VDR	-8.81	-10.51	-10.46	0.05

unseen molecules.

4.2. Projecting Molecules Generated by Structure-Based Drug Design Models

Generative models for structure-based drug design (SBDD) build molecules to fit a protein pocket structure (Luo et al., 2021). SBDD models often produce chemically invalid structures, because they operate in the 3D space with a few chemical constraints (Harris et al., 2023). As a remedy, we apply our model to molecules proposed by the SBDD models to find valid and synthesizable analogs.

We use Pocket2Mol (Peng et al., 2022) to generate molecules for targets in the LIT-PCBA dataset (Tran-Nguyen et al., 2020). The LIT-PCBA dataset contains 15 protein targets, each of which has at least one X-ray structure that can be used to run Pocket2Mol to generate molecules. For each target, we select 300 generated molecules based on the sum of QED and SA scores. Then, we apply our model to generate 5 analogs for each molecule and select best one according to Vina scores (Eberhardt et al., 2021). The estimated binding energy in kcal/mol shows slight improvement

for the majority of targets (Table 3), suggesting that the top-1 synthesizable analogs of each molecule could maintain binding. Figure 4 presents two examples in which the synthesizable analogs improve predicted binding. Figure 6 and 7 in the appendix present more examples.

4.3. Projecting Molecules Generated by Goal-Directed Generative Models

Goal-directed generative models are pivotal in molecular design, yet their practicality is often hampered by the synthesizability of the produced molecules. This issue was highlighted by Gao & Coley, who, utilizing three algorithms on the GuacaMol benchmark suite's multi-property objectives (Brown et al., 2019), revealed that around 70% of the molecules generated were considered unsynthesizable by the ASKCOS retrosynthesis analysis tool (Coley et al., 2019b).

We applied our model to these molecules which were labeled as "unsynthesizable" by ASKCOS. The analogs show limited structural similarity with an average Morgan fingerprint similarity score of 0.43 (Table 4). This divergence

Table 4. In general, the molecules generated by goal-directed generative models do not show significant similarity to their analogs. The projection typically results in a drop in the objective scores. Nonetheless, the maximum scores of the analogs on 6 targets decrease by an amount of less than 0.1, indicating that the properties are still preserved for some molecules.

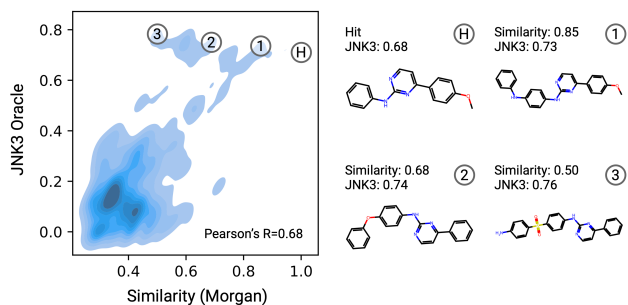| Property	Sim. Morgan	Sim. Scaffold	Sim. Gobbi	Avg(Objective)			Max(Objective)		
				Gen.	Analog	Δ	Gen.	Analog	Δ
Amlodipine MPO	0.4593	0.3098	0.3855	0.8110	0.4824	-0.3286	0.8895	0.8793	-0.0102
Deco Hop	0.5499	0.8273	0.7173	0.9768	0.7296	-0.2472	0.9992	0.9652	-0.0340
Fexofenadine MPO	0.4510	0.4589	0.3431	0.9316	0.4321	-0.4995	1.0000	0.8999	-0.1001
Osimertinib MPO	0.4339	0.3956	0.5620	0.9030	0.5424	-0.3606	0.9508	0.9326	-0.0182
Perindopril MPO	0.3175	0.2605	0.3181	0.7169	0.3552	-0.3617	0.7850	0.7011	-0.0839
Ranolazine MPO	0.4145	0.4648	0.5029	0.8889	0.5652	-0.3237	0.9103	0.8377	-0.0726
Scaffold Hop	0.5092	0.7499	0.6125	0.9558	0.5283	-0.4275	1.0000	0.8993	-0.1007
Sitagliptin MPO	0.2536	0.2230	0.2427	0.6527	0.0158	-0.6369	0.8437	0.4909	-0.3528
Valsartan SMARTS	0.3500	0.3558	0.2572	0.8191	0.0036	-0.8155	0.9861	0.4776	-0.5085
Zaleplon MPO	0.5328	0.5653	0.5429	0.6379	0.2502	-0.3877	0.7151	0.7122	-0.0029

Figure 5. The distribution of JNK3 oracle scores and similarity of analogs generated by expanding the hit molecule.

in structure is primarily due to the non-synthesizability of the original molecules, making it hard to compose a similar analog using the provided building blocks and reactions of the chemical space. The property scores achieved by the original unsynthesizable molecules are affected by the structural discrepancies, as indicated by the drop in average objective score. However, for most of the targets, there still exists at least one analog that mostly retains the original property as shown by the maximum objective scores.

In conclusion, the projection generally leads to synthesizable analogs that are sub-optimal in terms of the oracle property scores. These sub-optimal analogs share some common structural factors with the original molecules, hopefully making them good starting points for combinatorial optimization within the chemical space (Gao et al., 2021; Swanson et al., 2023).

4.4. Projecting Molecules to Explore Local Chemical Space for Hit Expansion

Our model's capabilities also extend to hit expansion in drug discovery, offering a strategic approach to identifying structurally similar, synthesizable analogs of hit molecules (Keserű & Makara, 2006; Levin et al., 2023).

We showcase our model's hit expansion capability by designing inhibitors for c-Jun N-terminal Kinases-3 (JNK3), following the setting of Levin et al.. Based on the oracle function provided by Li et al., our goal is to identify high-scoring, synthesizable molecules similar to a specified hit molecule with a JNK3 score of 0.68. We generated 500 analogs for the hit, resulting in a diverse array of analogs with different JNK3 scores and structural similarities, as depicted in Figure 5. Remarkably, 21 analogs outperformed the original hit in JNK3 score while maintaining an average Tanimoto similarity of 0.67. Additionally, these analogs present distinct synthetic pathways, different from traditional synthesis planning and building block exchange strategies (Levin et al., 2023), underlining our model's capability to cover broader chemical space (Figure 8 in the appendix).

5. Conclusion

In this work, we introduce a method that projects molecules proposed by generative models into synthesizable chemical space. The method is composed of a novel postfix notation-based representation of synthesis and a transformer-based encoder-decoder network. We study the application of the method in structure-based drug design, goal-directed molecular generation, and hit expansion. The results demonstrate that the model can effectively convert the generated molecules into synthesizable analogs while maintaining the desired properties. As opposed to previous methods that rely on combinatorial optimization or virtual screening, the proposed approach enables efficient exploration of the chemical space by utilizing the flexibility of generative models while respecting synthetic feasibility, further unlocking the potential of machine learning in molecular design.

Acknowledgement

Impact Statement

This paper presents work whose goal is to advance the generative modeling of molecular structures. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Bohacek, R. S., McMartin, C., and Guida, W. C. The art and practice of structure-based drug design: a molecular modeling perspective. *Medicinal research reviews*, 16(1): 3–50, 1996.

Bradshaw, J., Paige, B., Kusner, M. J., Segler, M., and Hernández-Lobato, J. M. A model to search for synthesizable molecules. *Advances in Neural Information Processing Systems*, 32, 2019.

Bradshaw, J., Paige, B., Kusner, M. J., Segler, M., and Hernández-Lobato, J. M. Barking up the right tree: an approach to search over molecule synthesis dags. *Advances in neural information processing systems*, 33:6852–6866, 2020.

Brown, N., Fiscato, M., Segler, M. H., and Vaucher, A. C. Guacamol: benchmarking models for de novo molecular design. *Journal of chemical information and modeling*, 59(3):1096–1108, 2019.

Button, A., Merk, D., Hiss, J. A., and Schneider, G. Automated de novo molecular design by hybrid machine intelligence and rule-driven chemical synthesis. *Nature machine intelligence*, 1(7):307–315, 2019.

Coley, C. W., Jin, W., Rogers, L., Jamison, T. F., Jaakkola, T. S., Green, W. H., Barzilay, R., and Jensen, K. F. A graph-convolutional neural network model for the prediction of chemical reactivity. *Chemical science*, 10(2): 370–377, 2019a.

Coley, C. W., Thomas III, D. A., Lummiss, J. A., Jaworski, J. N., Breen, C. P., Schultz, V., Hart, T., Fishman, J. S., Rogers, L., Gao, H., et al. A robotic platform for flow synthesis of organic compounds informed by ai planning. *Science*, 365(6453):eaax1566, 2019b.

Daylight Chemical Information Systems, Inc. SMARTS - A Language for Describing Molecular Patterns. https://www.daylight.com/dayhtml/doc/theory/theory.smarts.html, 2023. Accessed: Jan. 2024.

Eberhardt, J., Santos-Martins, D., Tillack, A. F., and Forli, S. Autodock vina 1.2. 0: New docking methods, expanded force field, and python bindings. *Journal of chemical information and modeling*, 61(8):3891–3898, 2021.

Enamine. Building blocks catalog. https://enamine.net/building-blocks/building-blocks-catalog, 2023. [Accessed 01-10-2023].

Ertl, P. and Schuffenhauer, A. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of cheminformatics*, 1:1–11, 2009.

Fu, T., Gao, W., Xiao, C., Yasonik, J., Coley, C. W., and Sun, J. Differentiable scaffolding tree for molecule optimization. In *International Conference on Learning Representations*, 2021.

Fu, T., Gao, W., Coley, C., and Sun, J. Reinforced genetic algorithm for structure-based drug design. *Advances in Neural Information Processing Systems*, 35:12325–12338, 2022.

Gao, W. and Coley, C. W. The synthesizability of molecules proposed by generative models. *Journal of chemical information and modeling*, 60(12):5714–5723, 2020.

Gao, W., Mercado, R., and Coley, C. W. Amortized tree generation for bottom-up synthesis planning and synthesizable molecular design. In *International Conference on Learning Representations*, 2021.

Gao, W., Fu, T., Sun, J., and Coley, C. Sample efficiency matters: a benchmark for practical molecular optimization. *Advances in Neural Information Processing Systems*, 35:21342–21357, 2022.

Gaulton, A., Bellis, L. J., Bento, A. P., Chambers, J., Davies, M., Hersey, A., Light, Y., McGlinchey, S., Michalovich, D., Al-Lazikani, B., et al. Chembl: a large-scale bioactivity database for drug discovery. *Nucleic acids research*, 40(D1):D1100–D1107, 2012.

Gobbi, A. and Poppinger, D. Genetic optimization of combinatorial libraries. *Biotechnology and bioengineering*, 61(1):47–54, 1998.

Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams,

R. P., and Aspuru-Guzik, A. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

Gottipati, S. K., Sattarov, B., Niu, S., Pathak, Y., Wei, H., Liu, S., Blackburn, S., Thomas, K., Coley, C., Tang, J., et al. Learning to navigate the synthetically accessible chemical space using reinforcement learning. In *International conference on machine learning*, pp. 3668–3679. PMLR, 2020.

Harris, C., Didi, K., Jamasb, A. R., Joshi, C. K., Mathis, S. V., Lio, P., and Blundell, T. Benchmarking generated poses: How rational is structure-based drug design with generative models? *arXiv preprint arXiv:2308.07413*, 2023.

Hartenfeller, M. and Schneider, G. De novo drug design. *Chemoinformatics and computational chemical biology*, pp. 299–323, 2011.

Hartenfeller, M., Zettl, H., Walter, M., Rupp, M., Reisen, F., Proschak, E., Weggen, S., Stark, H., and Schneider, G. Dogs: reaction-driven de novo design of bioactive compounds. *PLoS computational biology*, 8(2):e1002380, 2012.

Horwood, J. and Noutahi, E. Molecular design in synthetically accessible chemical space via deep reinforcement learning. *ACS omega*, 5(51):32984–32994, 2020.

Janela, T. and Bajorath, J. Simple nearest-neighbour analysis meets the accuracy of compound potency predictions using complex machine learning models. *Nature Machine Intelligence*, 4(12):1246–1255, 2022.

Jensen, J. H. 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.

Jin, W., Barzilay, R., and Jaakkola, T. Junction tree variational autoencoder for molecular graph generation. In *International conference on machine learning*, pp. 2323–2332. PMLR, 2018.

Keserű, G. M. and Makara, G. M. Hit discovery and hit-to-lead approaches. *Drug discovery today*, 11(15-16):741–748, 2006.

Kim, S., Chen, J., Cheng, T., Gindulyte, A., He, J., He, S., Li, Q., Shoemaker, B. A., Thiessen, P. A., Yu, B., et al. Pubchem 2023 update. *Nucleic acids research*, 51(D1):D1373–D1380, 2023.

Korovina, K., Xu, S., Kandasamy, K., Neiswanger, W., Poczos, B., Schneider, J., and Xing, E. Chembo: Bayesian optimization of small organic molecules with synthesizable

recommendations. In *International Conference on Artificial Intelligence and Statistics*, pp. 3393–3403. PMLR, 2020.

Levin, I., Fortunato, M. E., Tan, K. L., and Coley, C. W. Computer-aided evaluation and exploration of chemical spaces constrained by reaction pathways. *AIChE journal*, 69(12):e18234, 2023.

Li, Y., Zhang, L., and Liu, Z. Multi-objective de novo drug design with conditional graph generative model. *Journal of cheminformatics*, 10(1):1–24, 2018.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Luo, S., Guan, J., Ma, J., and Peng, J. A 3d generative model for structure-based drug design. *Advances in Neural Information Processing Systems*, 34:6229–6239, 2021.

Lyu, J., Wang, S., Balius, T. E., Singh, I., Levit, A., Moroz, Y. S., O'Meara, M. J., Che, T., Algaa, E., Tolmachova, K., et al. Ultra-large library docking for discovering new chemotypes. *Nature*, 566(7743):224–229, 2019.

Morgan, H. L. The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service. *Journal of chemical documentation*, 5(2):107–113, 1965.

Peng, X., Luo, S., Guan, J., Xie, Q., Peng, J., and Ma, J. Pocket2mol: Efficient molecular sampling based on 3d protein pockets. In *International Conference on Machine Learning*, pp. 17644–17655. PMLR, 2022.

Schwaller, P., Laino, T., Gaudin, T., Bolgar, P., Hunter, C. A., Bekas, C., and Lee, A. A. Molecular transformer: a model for uncertainty-calibrated chemical reaction prediction. *ACS central science*, 5(9):1572–1583, 2019.

Shoichet, B. K. Virtual screening of chemical libraries. *Nature*, 432(7019):862–865, 2004.

Swanson, K., Liu, G., Catacutan, D., Zou, J., and Stokes, J. Generative ai for designing and validating easily synthesizable and structurally novel antibiotics. In *NeurIPS 2023 Generative AI and Biology (GenBio) Workshop*, 2023.

Thakkar, A., Chadimová, V., Bjerrum, E. J., Engkvist, O., and Reymond, J.-L. Retrosynthetic accessibility score (rascore)–rapid machine learned synthesizability classification from ai driven retrosynthetic planning. *Chemical Science*, 12(9):3339–3349, 2021.

Tran-Nguyen, V.-K., Jacquemard, C., and Rognan, D. Lit-pcba: an unbiased data set for machine learning and virtual screening. *Journal of chemical information and modeling*, 60(9):4263–4273, 2020.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Venkatasubramanian, V., Chan, K., and Caruthers, J. M. Evolutionary design of molecules with desired properties using the genetic algorithm. *Journal of Chemical Information and Computer Sciences*, 35(2):188–195, 1995.

Vinkers, H. M., de Jonge, M. R., Daeyaert, F. F., Heeres, J., Koymans, L. M., van Lenthe, J. H., Lewi, P. J., Timmerman, H., Van Aken, K., and Janssen, P. A. Synopsis: synthesize and optimize system in silico. *Journal of medicinal chemistry*, 46(13):2765–2773, 2003.

Voršilák, M., Kolář, M., Čmelo, I., and Svozil, D. Syba: Bayesian estimation of synthetic accessibility of organic compounds. *Journal of cheminformatics*, 12(1):1–13, 2020.

Wouters, O. J., McKee, M., and Luyten, J. Estimated research and development investment needed to bring a new medicine to market, 2009-2018. *Jama*, 323(9):844–853, 2020.

Ying, C., Cai, T., Luo, S., Zheng, S., Ke, G., He, D., Shen, Y., and Liu, T.-Y. Do transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems*, 34:28877–28888, 2021.

You, J., Liu, B., Ying, Z., Pande, V., and Leskovec, J. Graph convolutional policy network for goal-directed molecular graph generation. *Advances in neural information processing systems*, 31, 2018.

A. Implementation

A.1. Training Data

The algorithm for the construction of (postfix notation, molecular graph) pairs is described below:

Algorithm 1 Construct a (postfix notation, molecular graph) pair for training

---

    Input: m_r, maximum number of reactions
    Input: m_a, maximum number of atoms
    Input: $\mathcal{F} : M \mapsto \{R\}$, a function that returns available reactions for molecule M. When $M = $ `null`, it returns all the reactions $\mathcal{R}$.
    Global: $\mathcal{H} : R \mapsto \{M\}$, a hash map that stores eligible molecules for each reaction.
    Initialize: $S \leftarrow []$ (empty stack)
    Initialize: $P \leftarrow []$ (empty postfix notation)
    while $\mathrm{CountReactions}(P) < m_r$ and $\mathrm{CountAtoms}(\mathrm{top}(S)) < m_a$ **do**
      Let $R \sim \mathcal{R}(\mathrm{top}(S))$ (randomly choose a reaction)
      Let $\{B_i : 1 \leq i \leq (\mathrm{NumReactants}(R) - \mathrm{len}(S))\} \sim \mathcal{H}(R)$ (randomly choose reactants for R)
      for $1 \leq i \leq (\mathrm{NumReactants}(R) - \mathrm{len}(S))$ **do**
        Push B_i onto S
        Append B_i to P
      end for
      Pop $\mathrm{NumReactants}(R)$ elements from S to **Call** R; Then **Push** the product Y onto S.
      Append R to P
    end while
    Return P, $\mathrm{top}(S)$ (postfix notation and its product molecular graph)

A.2. Parameters

The molecular graph encoder consists of 8 graph transformer layers, each of which has 8 attention heads, and the dimension of the input and output features is 512. The postfix notation decoder is a stack of 8 transformer decoder layers. Each has 8 attention heads, and the dimension of the input and output features is also 512. In the nearest-neighbor search, each building block molecule is indexed by the Morgan fingerprint (Morgan, 1965) of length 256 and radius 2. We use the AdamW optimizer (Loshchilov & Hutter, 2017) to train the network with a learning rate of 3e-4 and a batch size of 256 for 500,000 iterations. During the construction of synthesis for training, we limit the maximum number of reactions to 5 and the maximum number of atoms to 80. At inference time, we set the maximum sequence length to 16. If the end token is not generated within the limit, the process will be considered unsuccessful.

B. Additional Results

B.1. Stricter Test Split for Bottom-Up Synthesis Planning

To test the model's generalization more strictly, we construct a refined subset of the test set described in Section 4.1. Specifically, for each building block in the test set, we identify its most similar counterpart in the training set and calculate the maximum training set similarity accordingly. Building blocks with a maximum training set similarity greater than 0.6 are then removed, resulting in a refined set of test building blocks. We use this refined building block set to generate 1,000 molecules to test our model and SynNet, and the results are presented below.

Method	Success%	Recons.%	Sim.(Morgan)	Sim.(Scaffold)	Sim.(Gobbi)
SynNet	84.0%	10.0%	0.4552	0.6012	0.3348
Proposed	**95.8%**	**17.0%**	**0.5921**	**0.7239**	**0.6110**

Despite the stricter partition affecting the performance of both models, our model demonstrates superior generalization. Particularly noteworthy is the higher scaffold similarity, indicating that our model better preserves the scaffold of unseen molecules.

B.2. Baseline for Goal-Directed Generation

We present below a comparison between our model and the baseline model SynNet in the goal-directed generation setting. Our model outperforms the baseline in 7 out of 10 objectives in both similarity and property scores. Additionally, our model's performance is on par with the baseline in the remaining 3 objectives.

Property	Sim. Morgan		Sim. Scaffold		Sim. Gobbi		Avg(Obj)		Max(Obj)	
	SynNet	Ours	SynNet	Ours	SynNet	Ours	SynNet	Ours	SynNet	Ours
Amlodipine MPO	0.3732	**0.4593**	0.2333	**0.3098**	0.2787	**0.3855**	0.3358	**0.4824**	0.8605	**0.8793**
Deco Hop	0.4854	**0.5499**	0.5385	**0.8273**	0.5535	**0.7173**	0.6434	**0.7296**	0.9474	**0.9652**
Fexofenadine MPO	0.4308	**0.4510**	0.3942	**0.4589**	0.2983	**0.3431**	0.3501	**0.4321**	0.7928	**0.8999**
Osimertinib MPO	0.2375	**0.4339**	0.1740	**0.3956**	0.2494	**0.5620**	0.4827	**0.5424**	0.8613	**0.9326**
Perindopril MPO	**0.3184**	0.3175	0.2402	**0.2605**	0.2900	**0.3181**	0.2214	**0.3552**	0.6789	**0.7011**
Ranolazine MPO	0.3528	**0.4145**	0.2972	**0.4648**	0.2895	**0.5029**	0.3628	**0.5652**	**0.8451**	0.8377
Scaffold Hop	0.3694	**0.5092**	0.4965	**0.7499**	0.5082	**0.6125**	0.4593	**0.5283**	0.5299	**0.8993**
Sitagliptin MPO	**0.3309**	0.2536	**0.2852**	0.2230	**0.2711**	0.2427	**0.0302**	0.0158	0.4648	**0.4909**
Valsartan SMARTS	**0.3742**	0.3500	**0.3805**	0.3558	**0.2771**	0.2572	**0.0074**	0.0036	**0.6034**	0.4776
Zaleplon MPO	0.5141	**0.5328**	0.4884	**0.5653**	0.4856	**0.5429**	0.1477	**0.2502**	0.5548	**0.7122**

B.3. Baseline for Hit Expansion

In hit expansion, the objective is to identify similar molecules within the local chemical space. Given that random molecular edits may not guarantee synthesizability, we adopt querying a chemical database as the baseline method. Specifically, we search PubChem (Kim et al., 2023) using the hit molecule and retrieve the most similar 500 molecules based on fingerprint similarity. Only 7 out of the retrieved molecules surpass the hit in JNK3 score, with an average similarity score of 0.67. This result underscores the efficiency of our model in identifying relevant synthesizable compounds within the chemical space.

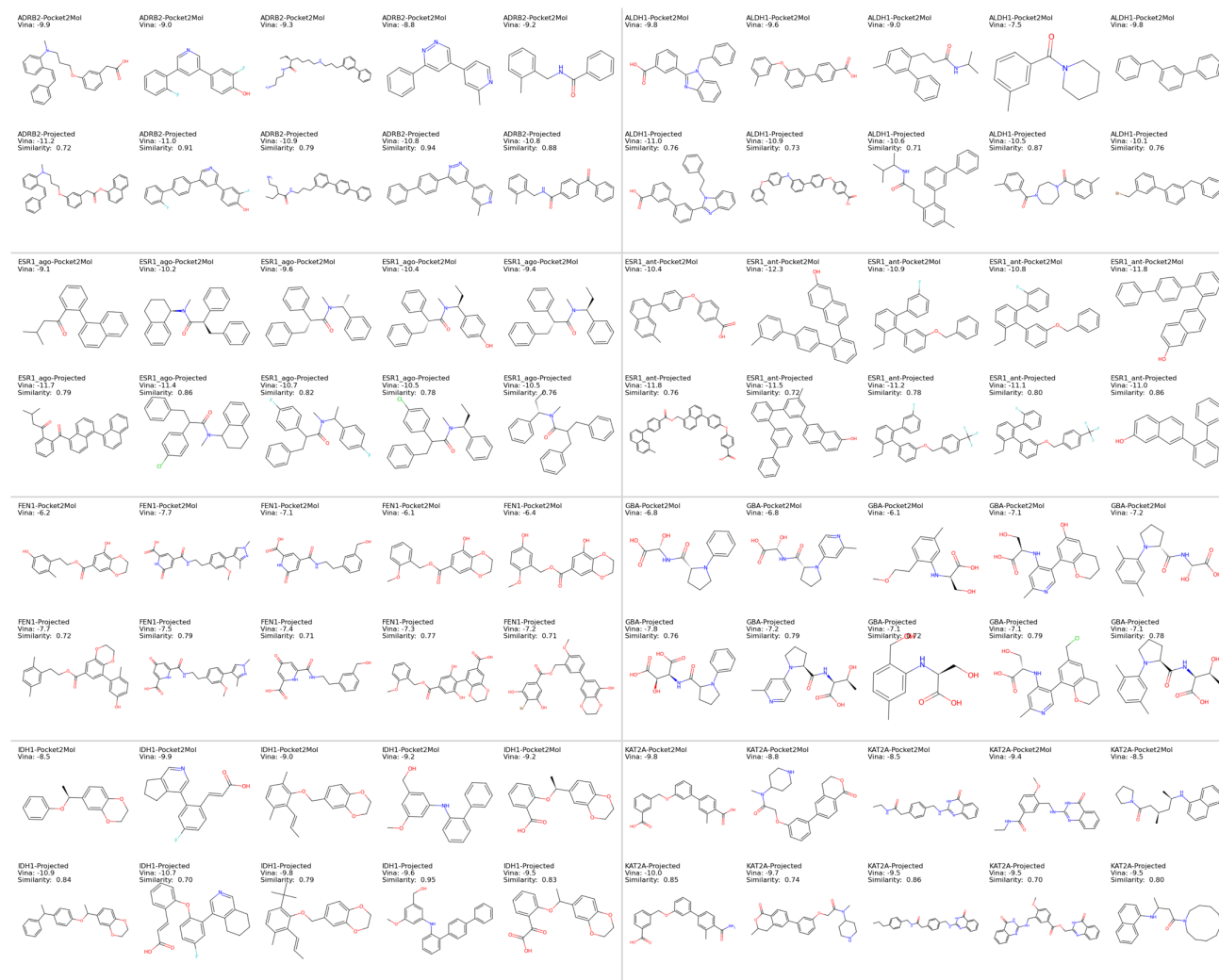## B.4. Molecules Generated by Pocket2Mol and Analogs

Figure 6. Molecules generated by Pocket2Mol and their analogs (1/2).
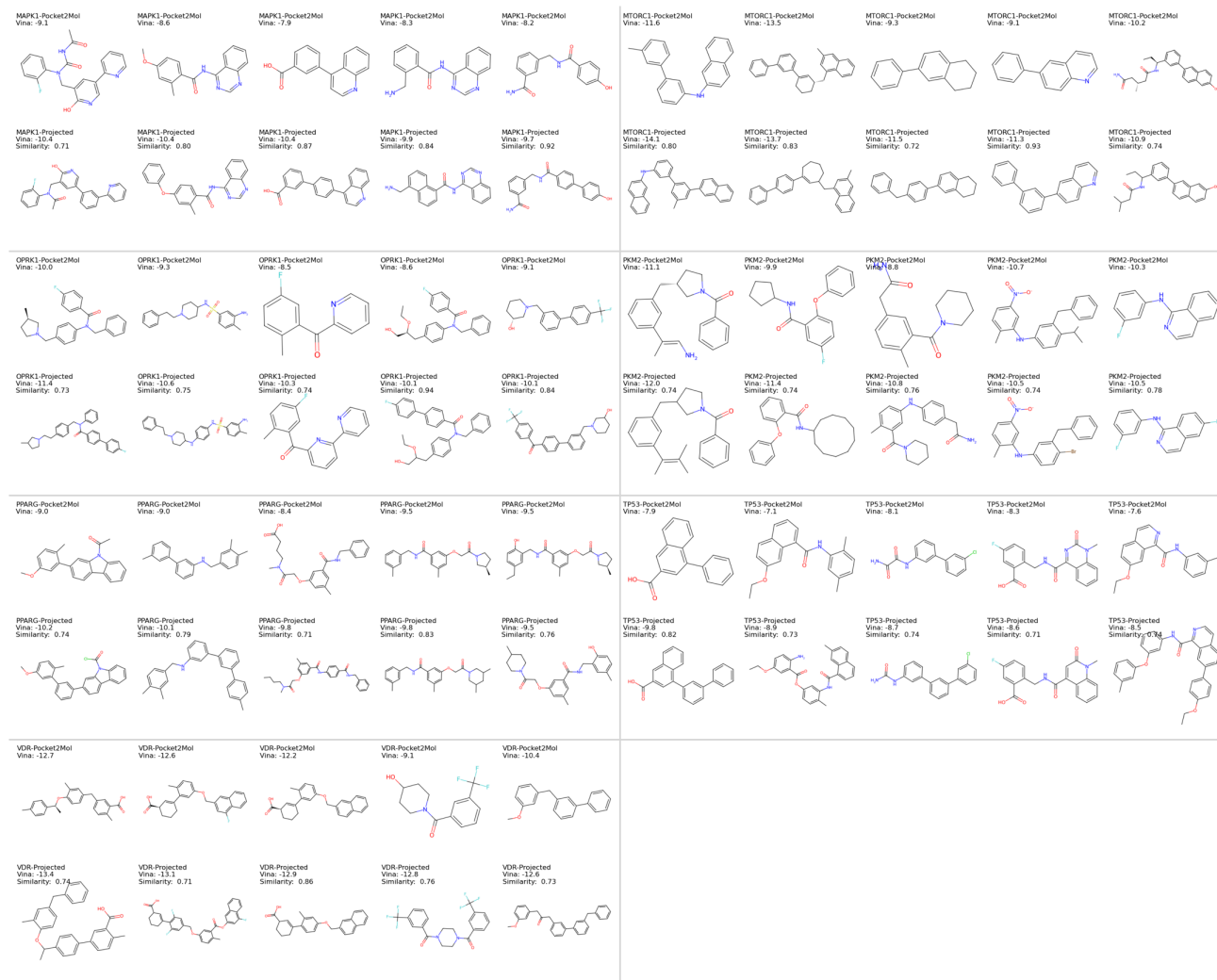
Figure 7. Molecules generated by Pocket2Mol and their analogs (2/2).

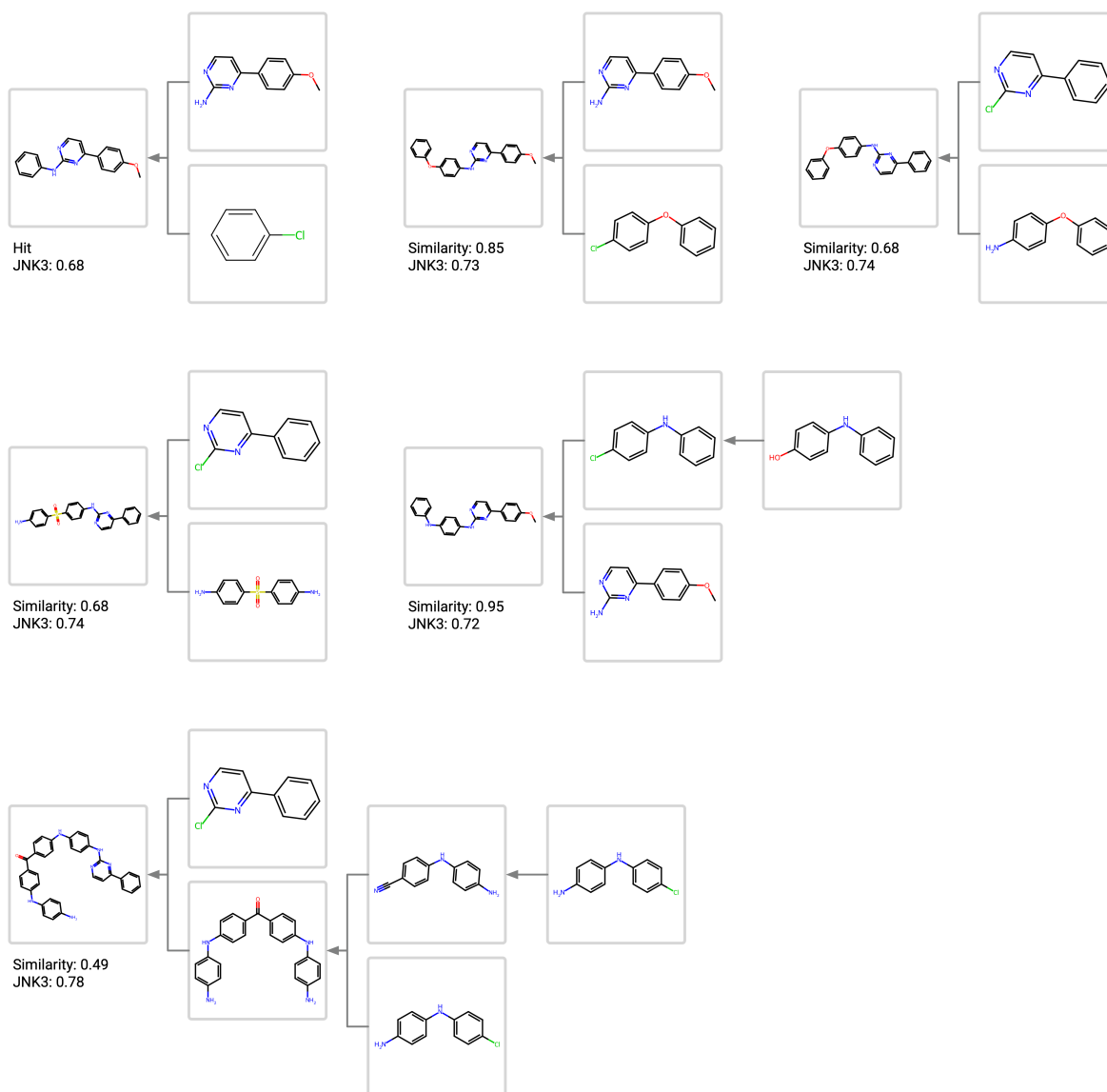## B.5. Synthetic Pathways in Hit Expansion

Figure 8. Distinct synthetic pathways of the analogs in hit expansion.