FragmentRetro: A Quadratic Retrosynthetic Method Based on Fragmentation Algorithms

Yu Shee, Anthony M. Smaldone, Anton Morgunov, Gregory W. Kyro, and Victor S. Batista*

Read Online

ACCESS | 📊 Metrics & More | 📖 Article Recommendations | 🆂🅸 Supporting Information
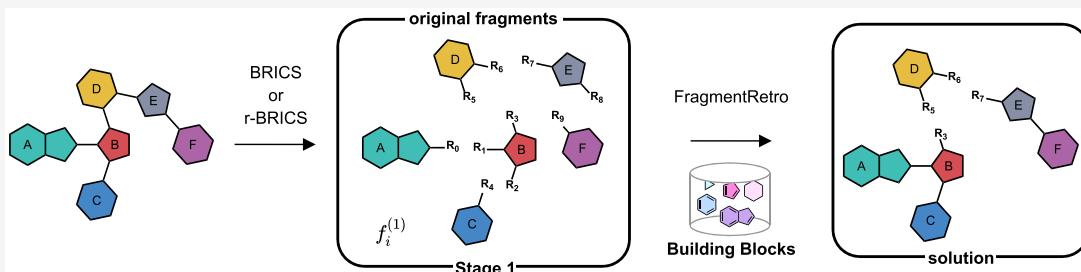
ABSTRACT: Retrosynthesis, the process of deconstructing a target molecule into simpler precursors, is crucial for computer-aided synthesis planning (CASP). Widely adopted tree-search methods often suffer from exponential computational complexity. In this work, we introduce FragmentRetro, a novel retrosynthetic method that leverages fragmentation algorithms, specifically BRICS and r-BRICS, combined with stock-aware exploration and pattern fingerprint screening to achieve quadratic complexity. FragmentRetro recursively combines molecular fragments and verifies their presence in a building block set, providing sets of fragment combinations as retrosynthetic solutions. We present the first formal computational analysis of retrosynthetic methods, showing that tree search exhibits exponential complexity $O(b^h)$, DirectMultiStep scales as $O(h^6)$, and FragmentRetro achieves $O(h^2)$, where h represents the number of heavy atoms in the target molecule and b is the branching factor for tree search. Evaluations on PaRoutes, USPTO-190, and natural products demonstrate that FragmentRetro achieves high solved rates with competitive runtime, including cases where tree search fails. The method benefits from fingerprint screening, which significantly reduces substructure matching complexity. While FragmentRetro focuses on efficiently identifying fragment-based solutions rather than full reaction pathways, its computational advantages and ability to generate strategic starting candidates establish it as a powerful foundational component for scalable and automated synthesis planning.

1. INTRODUCTION

Retrosynthetic analysis, the deconstruction of a target molecule into simpler, typically commercially available precursors, is fundamental to chemical synthesis and drug discovery.[1,2] Computer-aided synthesis planning (CASP) aims to automate this complex task. A conventional paradigm in CASP involves tree-search algorithms, such as Monte Carlo Tree Search (MCTS),[3] A* search (Retro*),[4] and Depth-First Proof Number (DFPN) search.[5] These methods iteratively apply single-step retrosynthetic (SSR) models, often machine learning-based,[6−9] to explore the vast combinatorial space of possible synthetic routes. While effective, a primary limitation of such tree-search approaches is their inherent exponential computational complexity with respect to tree depth, which can hinder scalability for large and complex targets. Recent advancements aim to mitigate this, for example, by incorporating graph-based neural networks,[10] reinforcement learning,[11] route context,[12] or higher-level abstractions of molecular features.[13] However, the underlying exponential scaling often persists in the worst-case scenario.

Alternative strategies have emerged to address these scalability challenges. DirectMultiStep (DMS) models, for instance, leverage sequence-to-sequence architectures to generate entire multistep pathways directly, bypassing iterative search.[14] While DMS has shown promising performance, particularly on benchmarks like PaRoutes,[15] and exhibits polynomial complexity, its $O(h^6)$ scaling with molecular size h can still be demanding. Other approaches, such as Double-Ended Synthesis Planning (DESP),[16] improve efficiency for specific use cases by conducting a bidirectional tree search when starting materials are provided by the user, though they still rely on iterative single-step predictions. This landscape motivates the exploration of fundamentally different algorithmic approaches
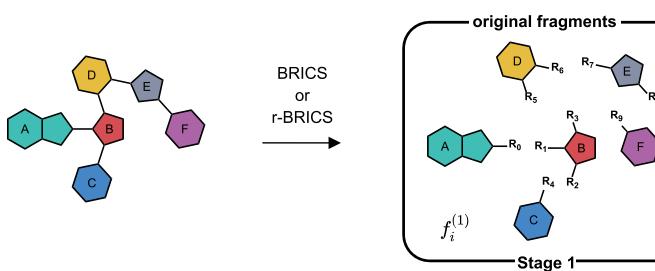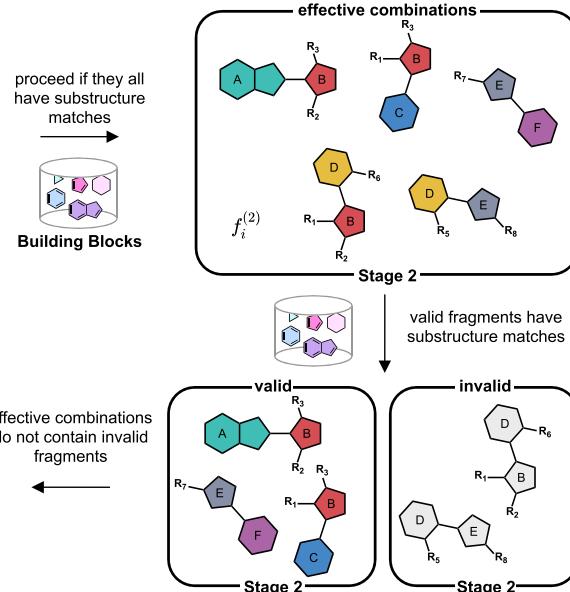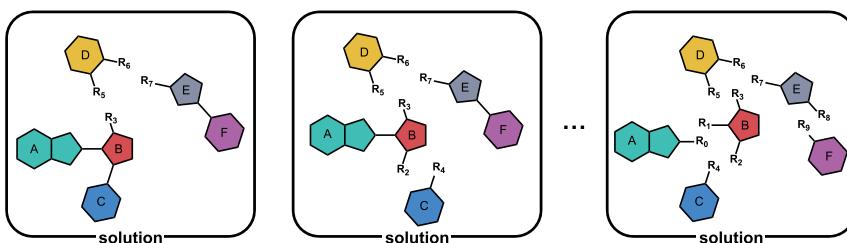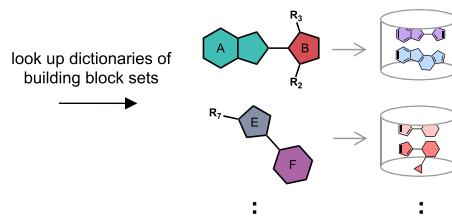
Figure 1. FragmentRetro algorithm. (a) Cartoon representation of an example molecule processed by BRICS or r-BRICS to yield initial fragments labeled A to F. (b) The FragmentRetro process: In Stage 1, all initial fragments have substructure matches in the stock set. In Stage 2, fragments A−B, B−C, and E−F are valid. In Stage 3, only fragment A−B−C remains valid. Fragments like A−B−D do not need to be checked, since B−D is invalid and therefore A−B−D cannot have a substructure match. There is no Stage 4, as no valid combinations of four fragments are possible in this case. (c) Possible solutions are sorted by the number of fragments, with the most efficient solution on the left. (d) Each valid fragment is associated with a subset of the stock that has substructure matches.

to retrosynthesis that offer improved computational efficiency, particularly for unconstrained initial exploration, without sacrificing solution quality.

In this work, we introduce FragmentRetro, a novel retrosynthetic method that adopts a bottom-up, fragment-based strategy (Figure 1). FragmentRetro utilizes molecular fragmentation algorithms (BRICS[17] and its revision, r-BRICS[18]) to decompose a target molecule. It then recursively combines these fragments, systematically verifying if the combined fragments exist as substructures within a provided set of commercially available building blocks (BBs). By integrating efficient substructure matching, augmented by pattern fingerprint and property screening, FragmentRetro identifies sets of fragment combinations that reconstruct the target. A key contribution of this paper is the first formal computational complexity analysis comparing these distinct retrosynthetic paradigms. We demonstrate that while tree-search methods scale exponentially ($O(b^h)$) and DMS polynomially ($O(h^6)$), FragmentRetro achieves a significantly more favorable quadratic

complexity ($O(h^2)$), albeit with a linear dependency on the stock set size which is effectively managed through parallelization.

We evaluate FragmentRetro on established benchmarks, including PaRoutes and USPTO-190, as well as challenging natural products. Our results show that FragmentRetro, particularly when using r-BRICS, achieves high solved rates, competitive with or exceeding those of tree-search and DMS methods in certain scenarios, often with substantially reduced runtimes. For instance, on USPTO-190 with a large BB set, FragmentRetro with r-BRICS attains the highest solved rate (78.4%) and demonstrates excellent parallelization for substructure matching. While FragmentRetro currently outputs sets of reconstructive fragments rather than fully elaborated reaction pathways in the form of directed acyclic graphs (DAGs), it provides a computationally efficient and scalable foundation for identifying viable precursor sets. This positions it as a powerful tool for initial synthesis exploration, potentially complementing other methods in a tiered approach to retrosynthesis. Our main contributions are (1) the FragmentRetro algorithm, a novel fragment-based retrosynthetic method with quadratic complex-

ity; (2) a comprehensive computational complexity analysis of major retrosynthetic paradigms; and (3) empirical validation demonstrating FragmentRetro's competitive performance and scalability.

2. RELATED WORK

The automation of retrosynthetic analysis has been a long-standing goal, evolving from rule-based systems to sophisticated machine learning-driven approaches. Understanding this landscape highlights the unique positioning of FragmentRetro.

2.1. Tree-Search Based Retrosynthesis

The predominant approach to CASP involves constructing a search tree or graph. Early systems relied on expert-defined chemical rules.[1] Modern methods typically integrate machine learning models for SSR predictions with advanced search algorithms. These SSR models can be template-based, utilizing predefined reaction patterns,[6−9,19−23] or template-free, directly predicting reactants from products.[24] Popular search algorithms include Monte Carlo Tree Search (MCTS),[3,25] A* search Retro*,[4] and Proof-Number Search derivatives like DFPN.[5]

Recent advancements in tree search focus on improving efficiency and prediction quality. For example, Xie et al. proposed RetroGraph, using graph neural networks to guide search on an explicit graph representation of intermediates to avoid redundant computations.[10] Reinforcement learning techniques, such as in GRASP,[11] train agents to learn optimal search policies. FusionRetro[12] aims to improve SSR predictions by incorporating contextual information from the partially built synthetic route. Roh et al. introduced a higher-level retrosynthesis strategy (Higherlev) that abstracts molecular details to simplify the search space, using substructure matching against BBs as a stopping criterion.[13] Despite these innovations, a fundamental challenge for tree-search methods is their worst-case exponential scaling with the size of the target molecule or the depth of the synthetic route, as formally analyzed in this work (SI Section B).

2.2. Direct Pathway Generation and Constrained Search

To circumvent the iterative nature and potential scalability issues of tree search, methods that directly generate entire multistep synthetic pathways have been developed. DirectMultiStep (DMS)[14] employs transformer-based sequence-to-sequence models to predict a full retrosynthetic route, represented as a linearized string, from a target molecule. DMS models have demonstrated strong performance on benchmarks like PaRoutes,[15] particularly in recovering known routes, and can incorporate constraints such as desired starting materials or route lengths. As shown in our complexity analysis (SI Section B), DMS exhibits polynomial complexity($O(h^6)$), offering an improvement over the exponential scaling of tree search.

Addressing the common real-world scenario where specific starting materials must be utilized, Yu et al. introduced Double-Ended Synthesis Planning (DESP).[16] DESP employs a bidirectional tree-search approach, simultaneously exploring retrosynthetic steps from the target and forward synthetic steps from user-specified starting materials. This method, guided by a learned synthetic distance cost function, has shown improved efficiency in solving such constrained problems compared to purely unidirectional search. However, like many tree-search methods, DESP combines explicit graph search with learned single-step predictors, and critically, it requires prior human specification of starting materials. Our work, in contrast, aims to

identify potential starting material sets automatically as part of its core fragment-based exploration.

2.3. Molecular Fragmentation in Chemistry

Molecular fragmentation, the process of breaking molecules into smaller, chemically meaningful pieces, is a well-established technique in cheminformatics. It finds applications in drug design for identifying pharmacophores, in quantitative structure–activity relationship (QSAR) modeling, and as a preprocessing step for various machine learning tasks. The BRICS (Breaking of Retrosynthetically Interesting Chemical Substructures) algorithm[17] is a widely used rule-based method that cleaves molecules at bonds deemed retrosynthetically labile, effectively identifying potential synthons or key disconnections. BRICS has been noted for sometimes producing large, inflexible fragments. To address this, r-BRICS (revised BRICS) was recently developed by Zhang et al., extending the original rule set to enable more granular fragmentation, particularly for challenging structures like fused rings and long aliphatic chains.[18] Our work, FragmentRetro, leverages these fragmentation algorithms not merely for analysis, but as the core engine for a bottom-up retrosynthetic search.

2.4. Substructure Searching

A critical component of FragmentRetro, as well as methods like Higherlev,[13] is substructure searching: determining if a given molecular fragment (query) exists within a larger molecule (often from a database of BBs). This is a subgraph isomorphism problem. Efficient algorithms like VF2[26] are commonly used in cheminformatics toolkits such as RDKit.[27] The practical performance of substructure searching can be significantly enhanced by prefiltering candidates using molecular fingerprints (e.g., pattern fingerprints) or simple property checks (e.g., heavy atom count, ring count), as employed in FragmentRetro. While worst-case complexity for subgraph isomorphism is exponential, these heuristics and optimized algorithms make it feasible for large-scale database searching in practice.[28]

3. ALGORITHM

FragmentRetro introduces a distinct paradigm for retrosynthetic analysis. Unlike top-down tree-search methods that recursively break bonds based on reaction predictions, or direct generation methods that learn entire pathway sequences, FragmentRetro operates bottom-up. It first decomposes the target into elementary fragments and then systematically explores combinations of these fragments that are present as substructures in a BB inventory. This stock-aware exploration of fragment combinations, combined with effective pruning and efficient screening, allows FragmentRetro to achieve quadratic computational complexity $O(h^2)$. While it does not directly output a sequence of reactions forming a DAG, it identifies sets of precursor fragments from which such a route could be constructed. This focus on computational efficiency and a novel search strategy distinguishes FragmentRetro from existing approaches, offering a scalable alternative for identifying potential synthetic BBs, particularly for complex targets or when exploring large chemical spaces. The formal complexity analysis presented herein further clarifies the theoretical advantages of this approach compared to established methods. In what follows, we define $\mathcal{M}$ as the set of all molecules and $\mathcal{B} \subset \mathcal{M}$ as the set of available BBs. For a formal statement of the retrosynthesis problem, refer to SI Section A.

3.1. Fragment-Based Retrosynthesis

Building upon the principles of molecular fragmentation discussed in the previous section, we introduce our fragment-based retrosynthetic approach that leverages substructure matching. This method recursively combines molecular fragments while verifying their presence in $\mathcal{B}$. The process aims to identify retrosynthetic solutions for a given target molecule $p^* \in \mathcal{M}$. Here, we define a retrosynthetic solution Q_i as a set of fragments that reconstruct the target molecule, i.e., $U_{f \in Q_i} f = p^*$ where the union is taken over molecular graphs with defined attachment points (fragmented bonds).

Let $\mathcal{F}_1 = \{f_1^{(1)}, f_2^{(1)}, ..., f_k^{(1)}\}$ represent the set of k molecular fragments obtained from a fragmentation method (e.g., BRICS[17] and r-BRICS[18]), where each fragment $f_i^{(1)} \in \mathcal{F}$ corresponds to a SMILES[29] string. These SMILES strings contain 'any' atoms (dummy atoms '*', which represent any atom in SMARTS[30] expressions) each attaching to the fragment bond ends unless no fragmentation is performed on a given SMILES string.

Fragment combinations are defined as groups of fragments that remain bonded during fragmentation, forming a connected subgraph of neighboring fragments. For example, $\mathcal{F}_1$ represents single fragments, while $\mathcal{F}_n$ represents combinations of n neighboring fragments, which are still represented as single SMILES strings, possibly with dummy atoms. These combinations expand as the recursion progresses, increasing the number of fragments combined and refining the search for valid retrosynthetic solutions. FragmentRetro identifies valid retrosynthetic solutions through the following recursive process:

1. **Initialization**: For each fragment $f_i^{(1)} \in \mathcal{F}_1$, check if $\exists\ b \in \mathcal{B}$ s.t. $f_i^{(1)}$ is a substructure of b, i.e., $f_i^{(1)} \subseteq b$. Here, strict substructure matches are done using SMARTS patterns to avoid branchings from nonfragmented sites. Also, the 'any' atoms can be hydrogen atoms (no branching from the fragmented sites). Aromaticity and chirality must match for a candidate BB to be considered as containing the substructure of the fragment, except at the fragmented sites where stereochemistry may be undefined. We call $f_i^{(1)}$ a valid fragment if there is a substructure match. If $\exists\ f_i^{(1)} \in \mathcal{F}_1$ s.t. $f_i^{(1)} \not\subseteq b \forall\ b \in \mathcal{B}$, the process terminates immediately, as subsequent fragment combinations would also fail to have substructure matches.

2. **Recursive Fragment Combination**: At each iteration $n \geq 2$, obtain combinations of n neighboring fragments (connected subgraphs with n nodes) from $\mathcal{F}_1$. Form a fragment set $\mathcal{F}_n = \{f_1^{(n)}, f_2^{(n)}, ..., f_{k^{(n)}}^{(n)}\}$, where each $f_i^{(n)}$ can be written as $f_{j_1}^{(1)} \cup ... \cup f_{j_n}^{(1)}$ with $j_1, ..., j_n \leq k$. For each fragment $f_i^{(n)} \in \mathcal{F}_n$, check if $\exists\ b \in \mathcal{B}$ such that $f_i^{(n)} \subseteq b$. We also record the subset of $\mathcal{B}$ that have a substructure match. When checking larger fragment combinations, we then take the intersection of these subsets, limiting the substructure check to a potentially much smaller set of BBs. One can save time by pruning fragments containing invalid fragments from previous stages to avoid redundant checks. This gives a subset of $\mathcal{F}_n$ that we call it the effective set $\mathcal{F}'_n$ at stage n. For example, if a combination $f_{j_1}^{(1)} \cup f_{j_2}^{(1)}$

at stage $n = 2$ does not have substructure matches in $\mathcal{B}$, any larger combination containing it (e.g., $f_{j_1}^{(1)} \cup f_{j_2}^{(1)} \cup f_{j_3}^{(1)}$) at stage $n = 3$ is excluded from evaluation). This, in practice, also prevents the process of finding all connected subgraphs with n nodes, as one can find the effective combinations from the valid combinations of the $n - 1$ stage. Additionally, strategies such as fingerprint screening can be used to filter out BBs that lack the pattern fingerprints of the fragment SMILES, significantly reducing the number of BBs that need to be checked for substructure matches (it is also applicable to the substructure matching process during the **Initialization** stage). In this work, FragmentRetro not only uses fingerprint screening but also utilizes screening based on the number of heavy atoms and rings, ensuring that BBs have greater than or equal numbers of heavy atoms and rings compared to the fragment combinations (i.e., property screening).

3. **Termination and Solution Construction**: The process terminates when either $\mathcal{F}'_n = \varnothing$ (indicating no effective or valid combinations exist) or $\mathcal{F}_n = \{p^*\}$ (or $n = |\mathcal{F}_1|$). The final retrosynthetic solutions $Q = \{Q_1, Q_2, ...\}$ consist of sets of fragments $Q_i = \{f_x^{(a)}, f_y^{(b)}, ...\}$, where each fragment $f_x^{(a)}$ represents a valid combination of original fragments from $\mathcal{F}_1$. Each Q_i satisfies the condition $U_{f \in Q_i} f = p^*$. The most efficient solution minimizes the number of fragments in Q_i, i.e., $|Q_i|$.

By iteratively expanding and pruning fragment combinations, FragmentRetro balances exhaustive exploration with computational efficiency, effectively navigating the constraints imposed by the BBs. This approach provides a robust framework for analyzing retrosynthetic pathways while deferring the selection of specific reactions to downstream analysis. The reaction selection process itself is relatively straightforward and depends heavily on the fragmentation algorithm. For instance, if the fragmentation algorithm (e.g., BRICS, as used in this work) is well-defined and consistent, it is possible to establish a mapping between specific fragment bond breaks and common reaction types associated with those bond breaks.

Unlike the explicit synthetic routes S generated in tree-search and DirectMultiStep methods, Q_i represents a set of fragments that reconstruct the target molecule p^* but does not inherently define the sequence of reaction steps. Without specifying the order of reactions, a DAG representation of the synthetic route cannot be constructed. For instance, given three fragments $f_x^{(a)}$, $f_y^{(b)}$, and $f_z^{(c)}$, it must be determined whether $f_x^{(a)}$ and $f_y^{(b)}$ react first, or if all three participate in a three-component reaction. Additionally, intermediate products formed during these reactions must be explicitly defined. Once the reaction order and intermediate products are specified, the sequence of reactions S can be constructed, enabling the formation of a DAG that represents the synthetic route.

3.2. Algorithm Summary

The FragmentRetro method employs a multistage, stock-aware exploration strategy for fragment combination. Each stage checks feasibility of fragment combinations, reducing the search space by pruning infeasible combinations. The algorithm is summarized in Algorithm 1 and an illustration is shown in Figure 1. Note that the effective combinations at stage n can be found by adding neighboring fragments to the valid combinations of

the $n − 1$ stage. This prevents the process of finding all connected subgraphs with n nodes.

Algorithm 1: FragmentRetro(p^*, $\mathcal{B}$, $\mathcal{F}_1 = \{f_1^{(1)}, f_2^{(1)}, \ldots, f_k^{(1)}\}$)

p^*: target molecule, $\mathcal{B}$: building blocks, $\mathcal{F}_1$: initial fragment set from fragmentation algorithms

for $i \leftarrow 1$ *to* $|\mathcal{F}_1|$ **do**
  if $f_i^{(1)} \not\subseteq b$ *for all* $b \in \mathcal{B}$ **then**
    exit ;   /* All initial fragments should be valid to continue */
  end
end
for $n \leftarrow 2$ *to* $|\mathcal{F}_1| − 1$ **do**
  Obtain $\mathcal{F}_n = \{f_1^{(n)}, f_2^{(n)}, \ldots, f_{k(n)}^{(n)}\}$;   /* n neighboring combinations */
  Prune to get effective combinations $\mathcal{F}'_n$;   /* Check for redundancy */
  if $\mathcal{F}'_n = \emptyset$ *or* $\mathcal{F}_n = \{p^*\}$ **then**
    /* No valid fragments exist or the target molecule is formed */
    break
  end
  if $f_i^{(n)} \subseteq b$ *for some* $b \in \mathcal{B}$ **then**
    /* Fingerprint screening on $\mathcal{B}$ can be performed here using $f_i^{(n)}$ */
    Mark $f_i^{(n)}$ as valid ;   /* Check substructure match in $\mathcal{B}$ */
  else
    Mark $f_i^{(n)}$ as invalid ;   /* Help check redundancy in the next stage */
  end
end
return solutions $Q = \{Q_1, Q_2, \ldots\}$, where $Q_i = \{f_x^{(a)}, f_y^{(b)}, \ldots\}$ satisfies $\bigcup_{f \in Q_i} f = p^*$

3.3. Computational Complexity Comparison

The computational complexity analysis is outlined in SI Section B and summarized in Table 1. Here, we use the heavy atom

Table 1. Comparison of Retrosynthetic Methods

method	complexity	feasibility guarantee	atom mapping	form DAG
tree search	$O(b^h)$	no	yes (template-based)	yes
DirectMultiStep	$O(h^6)$	no	no	yes
FragmentRetro	$O(h^2)$	no	yes	no

count of a given target h as a proxy for problem difficulty. For tree-search methods, the depth of the search tree for a given target can be estimated as $h/\Delta h$, where Δh is the average reduction in heavy-atom count per reaction. Therefore, tree-search methods scale exponentially as $O(b^{h/\Delta h}) = O(b^h)$, where b is the branching factor of the search tree and Δh can be treated as constant. In comparison, DirectMultiStep models require repeated computation of attention matrices and exhibit polynomial complexity of $O(h^6)$, since the route sequence

length scales as $O(h^2)$ and each attention operation scales as $O(h^4)$.

For FragmentRetro, the dominant computational cost arises from substructure matching. However, fingerprint screening eliminates the majority of nonmatching candidates before the substructure check, effectively keeping this process inexpensive in practice. This introduces a constant prefactor C_{fp} representing the filtering efficiency. The cost still scales linearly with the stock set size $|\mathcal{B}|$, and the number of possible fragment combinations scales quadratically with h. Therefore, the overall complexity of FragmentRetro can be derived as $O(C_{\text{fp}} \cdot |\mathcal{B}| \cdot h^2) = O(h^2)$. Note that this analysis describes the asymptotic behavior and comes with certain caveats. For example, we assume an ideal fingerprint that prevents collisions between molecular features, but in practice the fingerprint size is finite. Additionally, the worst case could occur for extremely ring-heavy target molecules, where r-BRICS fragmentation of ring bonds produces a large number of fragments and combinations, increasing the computational cost.

The exponential complexity of tree search also applies to the Higherlev approach,[13] which employs a higher-level retrosynthetic strategy by abstracting functional groups based on their electronegativity. For the Higherlev method, determining whether a precursor exists in the stock compound set is formulated as a substructure matching task. Since functional group abstraction increases Δh, their approach effectively reduces $h/\Delta h$ in practice. Consequently, following our derivation in SI Sections B.1 and B.3, the complexity of their method is given by $O(C_{\text{fp}} \cdot |\mathcal{B}| \cdot b^{h/\Delta h}) = O(b^h)$.

Table 1 provides a comparison of FragmentRetro, tree-search, and DMS methods across key metrics, including computational complexity, feasibility guarantees, and atom mapping capabilities. Notably, none of these methods can guarantee that the predicted synthetic routes are experimentally feasible. However, each method has certain advantages that may improve the likelihood of generating viable routes. **Tree-search methods** that rely on reaction templates allow them to retrieve references and metadata from template datasets, which can help assess reaction feasibility. **DirectMultiStep** employs a data-driven approach, generating reactions that are statistically closer to experimentally validated synthetic routes. **FragmentRetro** leverages fragmentation algorithms that preferentially break

Table 2. Search Performance on PaRoutes Test Sets with n_1 Stock or n_5 Stock as the Stock Set

method	targets	solved rate (%)	run time (s)[a]	clusters/solutions[b]
MCTS[c]	set-n_1	97.16	303.3	109
	set-n_5	96.89	365.7	113
Retro*[c]	set-n_1	97.28	300.7	31
	set-n_5	97.29	349.2	26
DFPN[c]	set-n_1	77.86	347.3	2
	set-n_5	67.30	297.9	2
DMS-Explorer-XL[d]	set-n_1	80.08	14.7	NA
	set-n_5	79.04	16.3	NA
FragmentRetro + BRICS[e]	set-n_1	69.90	9.4 (26.2)	4
	set-n_5	69.32	9.7 (27.1)	6
FragmentRetro + r-BRICS[e]	set-n_1	83.32	11.7 (30.2)	11
	set-n_5	82.62	12.0 (32.2)	12

[a]Averages over all targets. [b]Medians over all targets. NA indicates not available. [c]Data collected from the 2.0 version of PaRoutes in their GitHub repository[15] under the Apache-2.0 License. [d]DMS models are run with a beam size of 50 on a single NVIDIA A100 GPU with half-precision floating point inference (FP16). The other methods are run with a single CPU (no parallelization). [e]Runtimes in parentheses are reported without property and fingerprint screening.

bonds of retrosynthetic interest. Since these bonds are more likely to correspond to feasible reaction conditions, the resulting fragments can guide retrosynthetic planning effectively. Additionally, specific bond-breaking patterns can be mapped to known reaction types (similar to how synthons relate to actual molecules), suggesting that FragmentRetro could be extended to generate a DAG representation of synthetic routes. This remains an area for future development. Analogous to FragmentRetro's use of fragments, Higherlev also necessitates a mapping from abstracted portions to actual functional groups.

4. EMPIRICAL VALIDATION

4.1. Search Performance Evaluation

Table 2 presents the search performance of various methods on the PaRoutes test sets n_1 and n_5. All methods were evaluated using a single CPU/core (no parallelization), except for the DMS models, which were run on an NVIDIA A100 GPU. The stock set for each test follows the SMs defined in n_1 or n_5, as done in the PaRoutes work.[15] Among the evaluated methods, Retro*, a tree-search approach, achieved the highest solved rate. The DMS Explorer-XL model[14] exhibited competitive performance, solving a significant portion of the test set while providing a 20x speedup compared to tree-search methods. Similarly, FragmentRetro demonstrated decent performance, particularly with r-BRICS fragmentation, which outperformed DMS in solved rate while achieving a 30x speedup over tree-search methods. For FragmentRetro, a target is considered solved if at least one solution is found, where the solution corresponds to a set of valid fragments that can reconstruct the target compound, since the method does not output DAGs. Table 2 also reports the median number of solutions. Since the PaRoutes paper[15] clustered synthetic routes using the method in[32] to estimate route diversity, we compare the number of solutions from FragmentRetro to the number of clusters from PaRoutes. However, these metrics are not directly comparable, as a single FragmentRetro solution can correspond to multiple synthetic routes—each valid fragment combination may have substructure matches with different BBs.

Table 3 summarizes the search performance of different methods on the USPTO-190 dataset, using the Buyables stock set (0.329 M BBs) provided by the Higherlev study.[13] This stock set is a more practical choice than the eMolecule screening compounds (23.1 M compounds) used in Retro*,[4] as many screening compounds require custom synthesis. It is important to note that 13 compounds from USPTO-190 appear as target compounds in the single-step training dataset for Higherlev, while 47 compounds appear as targets or intermediates in the PaRoutes training set used by the DMS models. The DMS study reports solved rates both with and without these 47 compounds, showing minor differences. We expect similar results for Higherlev. FragmentRetro with BRICS fragmentation achieved a comparable solved rate to the best-performing DMS models but did not surpass Higherlev. However, with r-BRICS fragmentation, FragmentRetro achieved the highest solved rate among all methods. Despite this, the runtime of FragmentRetro is similar to that of tree-search methods in Table 2, as execution time scales with the stock set size ($|\mathcal{B}|$), even with the prefactor C_{fp}. This is due to substructure matching being the dominant computational bottleneck, as each candidate BB must be checked against the fragments. Fortunately, these checks are independent and can be efficiently parallelized across multiple CPU cores. To demonstrate this scalability, we also

Table 3. Search Performance on USPTO-190[a] with Buyables[b] as the Stock Set

method[c]	solved rate (%)	Run Time (s)[d]
DMS-Explorer-XL	27.9	21.9
Original + MCTS[13]	46.3	NA
DMS-Flash[e]	55.3	31.3
DMS-Wide[e]	56.8	105.8
Retro*-0[4f]	73.2	NA (65.0)
Higherlev + MCTS[13]	73.7	NA
Retro*[4f]	76.8	NA (57.1)
FragmentRetro + BRICS[g]	53.2	344.0 (74.2, 49.1, 42.1)
FragmentRetro + r-BRICS[g]	78.4	473.1 (102.1, 72.9, 64.5)

[a]USPTO-190 is from [4] (downloaded from [31] under the CC BY 4.0 License). [b]The stock compounds (Buyables) are from [13] under the CC BY 4.0 License and includes 0.329 M buyable building blocks from eMolecules, Sigma-Aldrich, Mcule, ChemBridge Hit2Lead, and WuXi LabNetwork. [c]DMS models are run with a beam size of 50 on a single NVIDIA A100 GPU with half-precision floating point inference (FP16). The other methods are run with a single CPU (no parallelization). [d]Averages over all targets. NA indicates that the runtime is not available in the original publication. [e]Uses step counts from 2 to 8 (total of 7 DMS model runs per compound). [f]Evaluated in this work using Retro*'s[4] official repository. The original checkpoint is used with a maximum of 500 iterations. Runtimes in parentheses are first-solution times, as the repository lacks search continuation. [g]Runtimes in parentheses are from parallelization with 5, 10, and 20 CPU cores (during substructure matching).

report FragmentRetro runtime using 5, 10, and 20 CPU cores. The results show that parallelization significantly reduces runtime, achieving speeds comparable to or better than DMS models. Notably, 5 and 10 CPU cores exhibit near-perfect parallelization efficiency, with speedups of 4.6× and 6.9×, respectively. However, with 20 CPU cores, the speedup is only 8.2x, indicating diminishing returns. This suggests that at this stock set size, the trade-off between parallelization efficiency and overhead is optimal around 10 CPU cores.

4.2. Case Studies

To demonstrate the practical utility of FragmentRetro, we evaluate drugs and natural products, as shown in Figure 2 (and SI Section C). These targets are selected from the Higherlev study[13] and represent different levels of retrosynthetic difficulty: Narlaprevir, which can be solved without a higher-level strategy; Martinellic Acid, which requires a higher-level strategy to be solved; and Lennoxamine, which is not solvable by either approach. For Narlaprevir, we apply BRICS fragmentation rules, while for Martinellic Acid and Lennoxamine, we use r-BRICS since these compounds contain fused rings that BRICS would otherwise leave intact. FragmentRetro generates hundreds of solutions for each of these compounds. Figure 2 presents a representative solution for each compound, selected to minimize the number of fragment combinations.

Each fragment combination can match multiple BBs; Figure 2 displays one representative building block per fragment combination. Some BBs are directly compatible for coupling, while others require additional preparation steps, such as functional group interconversion or protective group manipulation. As a result, the number of fragments in a solution does not directly correlate with the number of synthesis steps, similar to how higher-level abstraction methods operate.

977
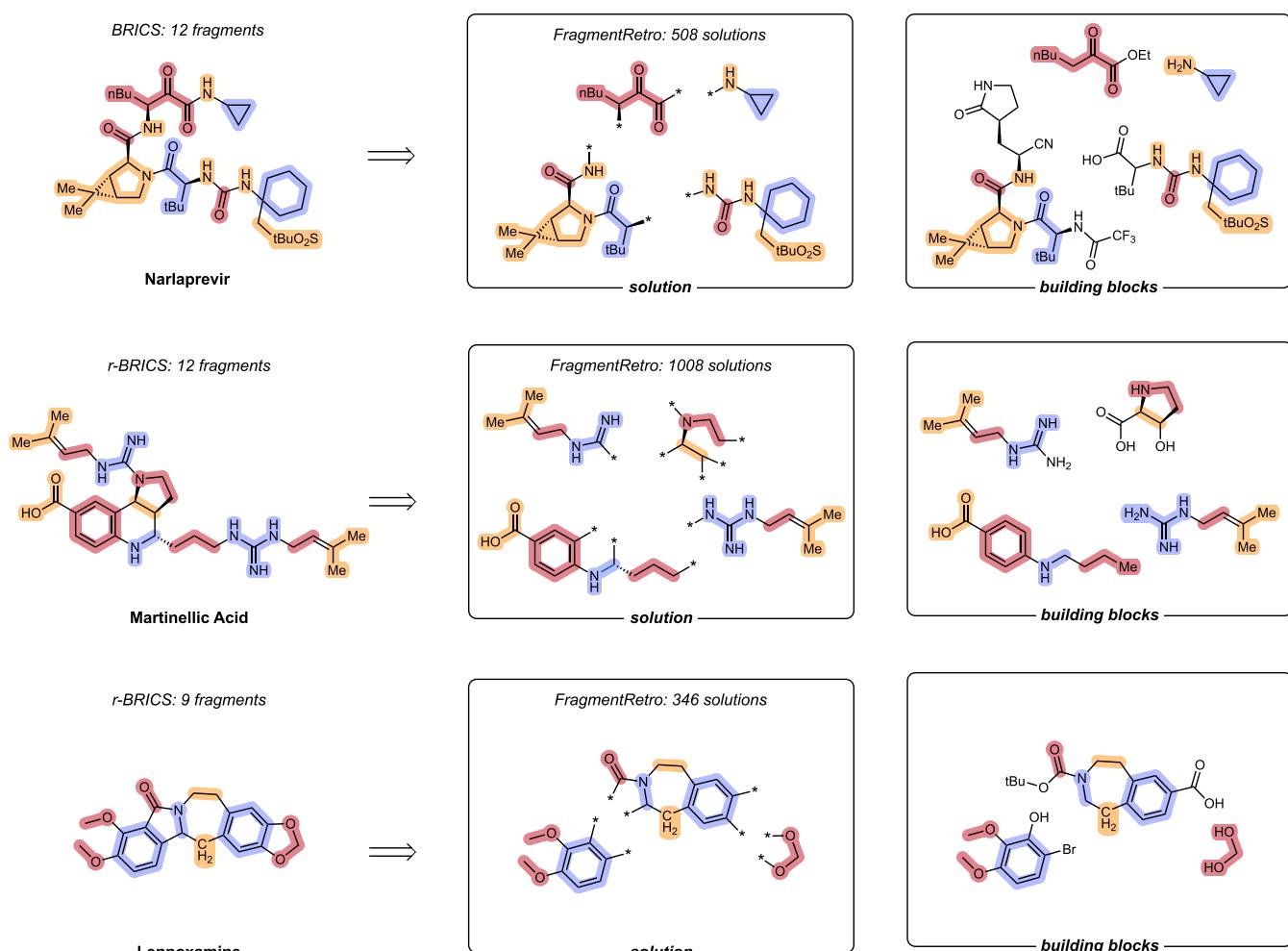
Figure 2. FragmentRetro evaluation on Narlaprevir, Martinellic Acid, and Lennoxamine. The fragments from BRICS and r-BRICS are highlighted. One solution for each compound from FragmentRetro is shown. The highlighted fragments remain highlighted in both the solutions and corresponding building blocks, even when some BRICS or r-BRICS cleavage sites are not fragmented in the solutions.

5. CONCLUSIONS

FragmentRetro provides a scalable framework for fragment-based retrosynthesis, achieving quadratic complexity through stock-aware fragment exploration and pruning. While FragmentRetro does not explicitly return a DAG, it performs the same core task as other retrosynthetic methods: identifying chemically valid decompositions of a target molecule into known precursors. Its outputs provide sets of fragment combinations that capture aspects of both single-step (individual disconnections) and multistep (multiple simultaneous combinations) approaches, although the method does not define the exact sequence of reactions or BBs. DAG construction is a post-processing step that depends on the fragment connectivity and can be layered atop our approach. Therefore, comparing the search complexity of FragmentRetro to tree-search and DMS methods is both appropriate and informative. Unlike DMS and tree-search methods that do not rely on template-based single-step models, FragmentRetro remains compatible with atom mapping tools, making it a flexible approach for retrosynthetic planning. However, FragmentRetro is not a route-level synthesis planning tool in its current form. Mapping solutions and BBs to fully elaborated synthetic routes (forming a DAG) would be required to enable direct route quality comparisons across different methods, potentially using top-k accuracy as an evaluation metric. Future work may focus on integrating cost-aware metrics and experimental validation to further refine its applicability.

This work also presents the first formal computational analysis of different retrosynthetic approaches. We establish that tree-search methods scale exponentially as $O(b^h)$, DMS has a polynomial complexity of $O(h^6)$, and FragmentRetro achieves $O(h^2)$ complexity, with the caveat that its runtime scales linearly with the stock set size. However, we show that substructure matching exhibits strong parallelization efficiency, particularly when applied to large building block datasets such as Buyables. Given the trade-offs between computational cost and solved rates, an efficient retrosynthesis pipeline could adopt a tiered approach: first applying FragmentRetro for rapid exploration, then leveraging DMS for higher route quality, and finally resorting to tree-search methods when higher solved rates are required. Overall, FragmentRetro represents a promising direction for fragment-based retrosynthesis, offering a computationally efficient option that complements existing methods.

ASSOCIATED CONTENT

Data Availability Statement

The data, code, and accompanying documentation for this work are available at https://github.com/randyshee/FragmentRetro

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jctc.5c01632.

> Problem formulation, computational complexity analysis, and additional case studies (PDF)

AUTHOR INFORMATION

Corresponding Author

Victor S. Batista − *Department of Chemistry, Yale University, New Haven, Connecticut 06520-8107, United States;* ⓘ orcid.org/0000-0002-3262-1237; Email: victor.batista@yale.edu

Authors

Yu Shee − *Department of Chemistry, Yale University, New Haven, Connecticut 06520-8107, United States;* ⓘ orcid.org/0000-0002-3728-0021

Anthony M. Smaldone − *Department of Chemistry, Yale University, New Haven, Connecticut 06520-8107, United States;* ⓘ orcid.org/0009-0008-7265-0017

Anton Morgunov − *Department of Chemistry, Yale University, New Haven, Connecticut 06520-8107, United States;* ⓘ orcid.org/0009-0004-6245-0354

Gregory W. Kyro − *Department of Chemistry, Yale University, New Haven, Connecticut 06520-8107, United States;* ⓘ orcid.org/0000-0002-0095-8548

Complete contact information is available at:
https://pubs.acs.org/10.1021/acs.jctc.5c01632

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

REFERENCES

(1) Corey, E. J.; Wipke, W. T. Computer-Assisted Design of Complex Organic Syntheses: Pathways for molecular synthesis can be devised with a computer and equipment for graphical communication. *Science* **1969**, *166*, 178−192.

(2) Szymkuć, S.; Gajewska, E. P.; Klucznik, T.; Molga, K.; Dittwald, P.; Startek, M.; Bajczyk, M.; Grzybowski, B. A. Computer-Assisted Synthetic Planning: The End of the Beginning. *Angew. Chem., Int. Ed.* **2016**, *55*, 5904−5937.

(3) Segler, M. H. S.; Preuss, M.; Waller, M. P. Planning Chemical Syntheses with deep neural networks and symbolic AI. *Nature* **2018**, *555*, 604−610.

(4) Chen, B.; Li, C.; Dai, H.; Song, L. *Retro*: Learning Retrosynthetic Planning with Neural Guided A* Search*; 37th International Conference on Machine Learning (ICML 2020); ICML, 2020.

(5) Kishimoto, A.; Buesser, B.; Chen, B.; Botea, A. *Depth-First Proof-Number Search with Heuristic Edge Cost and Application to Chemical Synthesis Planning*; Advances in Neural Information Processing Systems 32 (NeurIPS 2019); NIPS, 2019.

(6) Fortunato, M. E.; Coley, C. W.; Barnes, B. C.; Jensen, K. F. Data Augmentation and Pretraining for Template-Based Retrosynthetic Prediction in Computer-Aided Synthesis Planning. *J. Chem. Inf. Model.* **2020**, *60*, 3398−3407.

(7) Coley, C. W.; Rogers, L.; Green, W. H.; Jensen, K. F. Computer-Assisted Retrosynthesis Based on Molecular Similarity. *ACS Cent. Sci.* **2017**, *3*, 1237−1245.

(8) Ishida, S.; Terayama, K.; Kojima, R.; Takasu, K.; Okuno, Y. Prediction and interpretable visualization of retrosynthetic reactions using graph convolutional networks. *J. Chem. Inf. Model.* **2019**, *59*, 5026−5033.

(9) Shee, Y.; Li, H.; Zhang, P.; Nikolic, A. M.; Lu, W.; Kelly, H. R.; Manee, V.; Sreekumar, S.; Buono, F. G.; Song, J. J.; et al. Site-specific template generative approach for retrosynthetic planning. *Nat. Commun.* **2024**, *15*, No. 7818.

(10) Xie, S.; Yan, R.; Han, P.; Xia, Y.; Wu, L.; Guo, C.; Yang, B.; Qin, T.et al.*RetroGraph: Retrosynthetic Planning with Graph Search*; Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining; ACM: New York, NY, USA, 2022; pp 2120−2129.

(11) Yu, Y.; Wei, Y.; Kuang, K.; Huang, Z.; Yao, H.; Wu, F.*GRASP: Navigating Retrosynthetic Planning with Goal-driven Policy*; Advances in Neural Information Processing Systems; NIPS, 2022; pp 10257−10268.

(12) Liu, S.; Tu, Z.; Xu, M.; Zhang, Z.; Lin, L.; Ying, R.; Tang, J.; Zhao, P.; Wu, D.*FusionRetro: Molecule Representation Fusion Via In-context Learning for Retrosynthetic Planning*; International Conference on Machine Learning; ICML, 2023; pp 22028−22041.

(13) Roh, J.; Joung, J. F.; Yu, K.; Tu, Z.; Bartholomew, G. L.; Santiago-Reyes, O. A.; Fong, M. H.; Sarpong, R.; Reisman, S. E.; Coley, C. W. Higher-level Strategies for Computer-Aided Retrosynthesis *ChemRxiv* 2025.

(14) Shee, Y.; Morgunov, A.; Li, H.; Batista, V. S. DirectMultiStep: Direct Route Generation for Multistep Retrosynthesis. *J. Chem. Inf. Model.* **2025**, *65*, 3903−3914.

(15) Genheden, S.; Bjerrum, E. PaRoutes: towards a framework for benchmarking retrosynthesis route predictions. *Digital Discovery* **2022**, *1*, 527−539.

(16) Yu, K.; Roh, J.; Li, Z.; Gao, W.; Wang, R.; Coley, C. W. Double-Ended Synthesis Planning with Goal-Constrained Bidirectional Search. 2024. arXiv:2407.06334. arXiv.org e-Printarchive. https://doi.org/10.48550/arXiv.2407.06334.

(17) Degen, J.; Wegscheid-Gerlach, C.; Zaliani, A.; Rarey, M. On the art of compiling and using'drug-like'chemical fragment spaces. *ChemMedChem.* **2008**, *3*, 1503−1507.

(18) Zhang, L.; Rao, V.; Cornell, W. r-BRICS−A Revised BRICS Module That Breaks Ring Structures and Carbon Chains. *ChemMedChem* **2024**, *19*, No. e202300202.

(19) Coley, C.; Jin, W.; Rogers, L.; Jamison, T. F.; Jaakkola, T. S.; Green, W. H.; Barzilay, R.; Jensen, K. F. A graph-convolutional neural network model for the prediction of chemical reactivity. *Chem. Sci.* **2019**, *10*, 370−377.

(20) Seidl, P.; Renz, P.; Dyubankova, N.; Neves, P.; Verhoeven, J.; Wegner, J. K.; Segler, M.; Hochreiter, S.; Klambauer, G. Improving Few- and Zero-Shot Reaction Template Prediction Using Modern Hopfield Networks. *J. Chem. Inf. Model.* **2022**, *62*, 2111−2120.

(21) Segler, M. H. S.; Waller, M. P. Neural-Symbolic Machine Learning for Retrosynthesis and Reaction Prediction. *Chem. - Eur. J.* **2017**, *23*, 5966−5971.

(22) Chen, S.; Jung, Y. Deep retrosynthetic reaction prediction using local reactivity and global attention. *JACS Au* **2021**, *1*, 1612−1620.

(23) Coley, C. W.; Barzilay, R.; Jaakkola, T. S.; Green, W. H.; Jensen, K. F. Prediction of Organic Reaction Outcomes Using Machine Learning. *ACS Cent. Sci.* **2017**, *3*, 434−443.

(24) Jin, W.; Coley, C. W.; Barzilay, R.; Jaakkola, T. S.*Predicting Organic Reaction Outcomes with Weisfeiler-Lehman Network*; Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4−9; NIPS: Long Beach, CA, USA, 2017; pp 2607−2616.

(25) Genheden, S.; Thakkar, A.; Chadimová, V.; Reymond, J.-L.; Engkvist, O.; Bjerrum, E. AiZynthFinder: a fast, robust and flexible open-source software for retrosynthetic planning. *J. Cheminf.* **2020**, *12*, No. 70.

(26) Cordella, L. P.; Foggia, P.; Sansone, C.; Vento, M. *A (Sub) Graph Isomorphism Algorithm for Matching Large Graphs*; IEEE transactions on pattern analysis and machine intelligence; IEEE, 2004; pp 1367−1372 .

(27) Landrum, G. et al. *RDKit: Open-Source Cheminformatics*; RDKit, 2006.

(28) Ehrlich, H.-C.; Rarey, M. Systematic benchmark of substructure search in molecular graphs-From Ullmann to VF2. *J. Cheminf.* **2012**, *4*, No. 13.

(29) Daylight Theory: SMILES. https://www.daylight.com/dayhtml/doc/theory/theory.smiles.html. Accessed: March 10, 2025.

(30) Daylight Theory: SMARTS. https://www.daylight.com/dayhtml/doc/theory/theory.smarts.html. Accessed: March 10, 2025.

(31) Shee, Y.; Morgunov, A. Data for "DirectMultiStep: Direct Route Generation for Multistep Retrosynthesis" 2025 https://figshare.com/articles/dataset/Data_for_DirectMultiStep_Direct_Route_Generation_for_Multistep_Retrosynthesis_/28629470. ; Accessed: March 25, 2025.

(32) Genheden, S.; Engkvist, O.; Bjerrum, E. Fast prediction of distances between synthetic routes with deep learning. *Mach. Learn.: Sci. Technol.* **2022**, *3*, No. 015018.