It Takes Two to Tango: Directly Optimizing for *Constrained* Synthesizability in Generative Molecular Design

Anonymous authors
Paper under double-blind review

Abstract

Constrained synthesizability is an unaddressed challenge in generative molecular design. In particular, designing molecules satisfying multi-parameter optimization objectives, while simultaneously being synthesizable *and* enforcing the presence of specific building blocks in the synthesis. This is practically important for molecule re-purposing, sustainability, and efficiency. In this work, we propose a novel reward function called **TANimoto Group Overlap (TANGO)**, which uses chemistry principles to transform a sparse reward function into a *dense* reward function – crucial for reinforcement learning (RL). TANGO can augment molecular generative models to *directly* optimize for constrained synthesizability while simultaneously optimizing for other properties relevant to drug discovery. Our framework is general and addresses starting-material, intermediate, and divergent synthesis constraints. Contrary to many existing works in the field, we show that *incentivizing* a general-purpose model with RL is a productive approach to navigating challenging synthesizability optimization scenarios. We demonstrate this by showing that the trained models explicitly learn a desirable distribution. Our framework is the first *generative* approach to successfully address constrained synthesizability.

1 Introduction

Synthesizable generative molecular design is becoming increasingly prevalent (Gao & Coley, 2020; Stanley & Segler, 2023), paralleling the rise in the number of experimentally validated generative design case studies (Du et al., 2024). Controlling *how* generated molecules can be synthesized offers great potential for the push towards closed-loop discovery (Coley et al., 2020a;b) as molecules that can be made from specific reagents or reactions are naturally more amenable to robotic synthesis automation, which can be specialized for certain chemistries (Tom et al., 2024; Strieth-Kalthoff et al., 2024; Sin et al., 2024). Moving beyond methods that optimize for synthesizability heuristics (Stanley & Segler, 2023; Neeser et al., 2023), approaches that *explicitly* assess synthesizability can be broadly categorized into forward- or retro-synthesis which builds molecules from simple building blocks, or recursively decomposes a target molecule into constituent building blocks, respectively. An example of forward-synthesis in the context of molecular design is synthesizability-constrained molecular generation. These methods anchor molecular generation in viable chemical transformation rules, thus *promoting* synthesizability (Gao et al., 2022; 2024). On the other hand, retrosynthesis planning (Liu et al., 2017; Segler & Waller, 2017; Coley et al., 2017; Segler et al., 2018) proposes viable synthetic routes to a target molecule, and these models are often used as stand-alone tools to assess synthesizability. Such models have become increasingly adopted and are now routinely used to filter generated molecules (Shields et al., 2024). Recent work has shown that generative models can *directly* generate molecules deemed synthesizable by retrosynthesis models by treating them as another oracle (computational prediction) to optimize for (Guo & Schwaller, 2024c). Subsequently, *constrained* synthesis planning has become a research focus, whereby proposed synthetic routes incorporate *enforced building blocks*. This is especially relevant for sustainability and efficiency and examples include semi-synthesis (Vollmann et al., 2022) (start from reagents isolated from natural sources) and divergent-synthesis (Li et al., 2018) (pass through common intermediates). More examples include starting-material constrained synthesis (Granda et al., 2018; Wołos et al., 2020), which can also re-purpose waste to valuable molecules (Wołos et al., 2022; Żądło-Dobrowolska et al., 2024). More

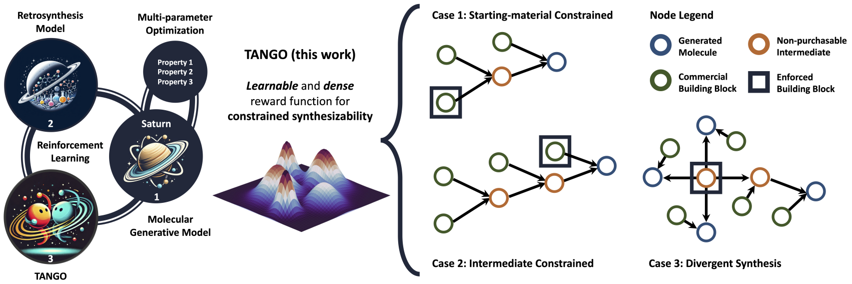

Retrosynthesis Model
Multi-parameter Optimization
Property 1
Property 2
Property 3
2
Saturn
Reinforcement Learning
1
Molecular Generative Model
3
TANGO
TANGO (this work)
Learnable and dense reward function for constrained synthesizability
Case 1: Starting-material Constrained
Node Legend
Generated Molecule
Non-purchasable Intermediate
Commercial Building Block
Enforced Building Block
Case 2: Intermediate Constrained
Case 3: Divergent Synthesis

Figure 1: TANGO guides the generation of molecules directly optimized for constrained synthesizability with *enforced building blocks* while simultaneously optimizing other properties. Our method generalizes across starting-material, intermediate, and divergent synthesis constraints.

recently, constrained retrosynthesis algorithms have been proposed (Johnson et al., 1992; Yu et al., 2022; 2024). However, to date, there are no molecular generative models that can enforce specific building blocks in the proposed routes.

In this work, we show that a *general-purpose* molecular generative model, without any constraints, can be *incentivized* to generate synthesizable molecules that satisfy multi-parameter optimization (MPO) objectives while jointly *enforcing* a set of building blocks. Our contribution is as follows: **(1)** We leverage chemistry principles and propose the **TANimoto Group Overlap (TANGO)** reward function to generate molecules deemed synthesizable by retrosynthesis models with the presence of *enforced building blocks* using reinforcement learning (RL). **(2)** We show that generated molecules satisfy MPO objectives, and by design, enable the construction of *synthesis networks* where *common intermediates* branch towards diverse, high-reward molecules. **(3)** We show that letting a *general purpose* model *freely learn* (using incentives), can be a productive approach to optimizing challenging synthesizability objectives.

2 Related Work

Retrosynthesis Models. Retrosynthesis planning aims to find a set of commercial building blocks and viable chemical transformations that can be combined to synthesize a target molecule. Existing works encode chemically plausible transformations either as reaction templates (coded patterns) (Chen & Jung, 2021; Xie et al., 2023) or template-free approaches (learn from data) operating on SMILES strings (Liu et al., 2017; Segler & Waller, 2017; Schwaller et al., 2020; Thakkar et al., 2023; Han et al., 2024) or graphs (Sacha et al., 2021; Zhong et al., 2023). Subsequently, multi-step retrosynthesis planning is tackled by coupling a search algorithm such as Monte Carlo tree search (Segler et al., 2018), Retro* (Chen et al., 2020), Planning with Dual Value Networks (PDVN) (Liu et al., 2023), or the recent Double-Ended Synthesis Planning (DESP) (Yu et al., 2024). With retrosynthesis planning being a ubiquitous task in molecular discovery, many platform solutions exist, including SYNTHIA (Szymkuć et al., 2016; Grzybowski et al., 2018), AiZynthFinder (Genheden et al., 2020; Saigiridharan et al., 2024), ASKCOS (Coley et al., 2019; Tu et al., 2025), Eli Lilly's LillyMol (Watson et al., 2019), Molecule.one's M1 platform (Molecule.one), and IBM RXN (Schwaller et al., 2020). In the context of generative molecular design, retrosynthesis models are usually used for post-hoc filtering due to their inference cost, but recent work has shown that with a sample-efficient model, they can be incorporated directly as an optimization objective (Guo & Schwaller, 2024c).

Synthesizability-constrained Molecular Generation. Bridging concepts from retrosynthesis, synthesizability-constrained models anchor molecular generation by enforcing a set of valid chemical transformations (Vinkers et al., 2003; Hartenfeller et al., 2012; Ghiandoni et al., 2022; 2024; Bradshaw et al., 2019; 2020; Korovina et al., 2020; Gao et al., 2022; Seo et al., 2023; Koziarski et al., 2024; Gao et al., 2024; Cretu et al., 2024; Luo et al., 2024; Gottipati et al., 2020; Horwood & Noutahi, 2020; Fialková et al., 2021; Jocys et al., 2024; Seo et al., 2024). To date, there are no molecular generative models that can enforce the presence of specific building blocks in the synthesis graph and the closest works are SynNet (Gao et al., 2022) and the very recent SynFormer (Gao et al., 2024) models which can condition on a target molecule to propose a synthetic route. Current synthesizability-constrained approaches cannot reliably (or are sample-inefficient) satisfy MPO objectives which is a necessary

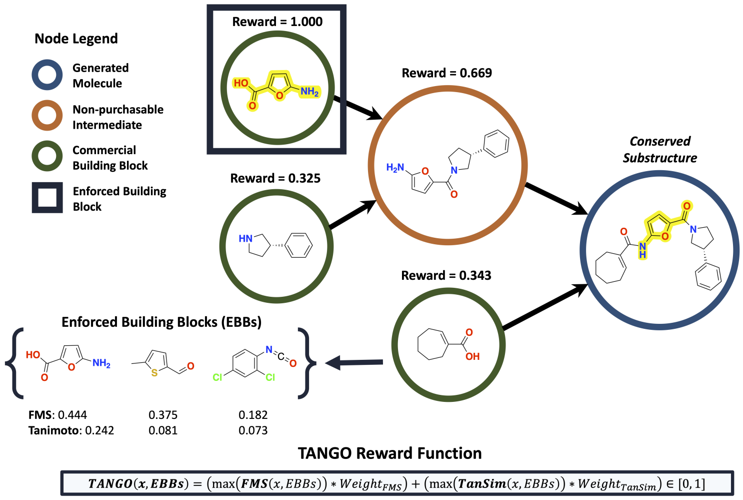


Figure 2: TANGO reward function: the maximum *similarity* between *every* non-root node (generated molecule) molecule and the set of enforced building blocks. Every synthesizable generated molecule returns a non-zero reward.

requirement for practical applications. In this work, we show that a general-purpose model, can generate synthesizable molecules that satisfy MPO objectives while *enforcing* the presence of a small set of building blocks either at the start of the synthesis (starting-material constrained), as a common intermediate (intermediate-constrained), or non-commercial building blocks that diverge to diverse, favorable generated molecules (divergent synthesis) (Fig. 1). To our knowledge, there are only several works Johnson et al. (1992); Yu et al. (2022; 2024); Szymkuć et al. (2016); Grzybowski et al. (2018) that enable some notion of building block-constrained synthesis planning. In particular, the very recent DESP (Yu et al., 2024) retrosynthesis search algorithm proposes a bidirectional search that can constrain on a starting-material. Our work differs in that we are not proposing a search algorithm, but rather the first *generative* approach that *jointly* tackles constrained synthesizability and MPO. Moreover, our framework can consider the constraint of *many* building blocks simultaneously.

3 Methods

In this section, we describe the problem formulation, the generative model, the **TANGO** reward function, and the experimental setup.

Constrained Synthesizability Problem Formulation. In synthesis planning, the goal is to propose a valid synthetic route to a target molecule using (commercially) available building blocks, B, and a set of reaction rules, R. We define a *synthesis graph*, $G(M, R)$, where each node represents an intermediate molecule, m, that need not necessarily be an available building block, b, and the edges represent reactions, $r \in R$. The *depth* of a node is the number of edges from the root node (the target molecule). A valid synthetic route requires that all leaf nodes correspond to commercially available building blocks, $b \in B$. We further define *enforced* building blocks, $B_{enf} \subseteq B$. In practice, $|B_{enf}| << |B|$, and in this work, we consider $|B_{enf}| \in \{10, 100\}$. We address three cases of constrained synthesis in this work:

Case 1: Starting-material Constrained Synthesis. A synthesis graph is considered *starting-material constrained* if at least one leaf node, $m \in G(M, R)$, satisfies both of the following conditions: (1) $m = b \in B_{enf}$, and (2) depth(m) = max depth:

$$\exists m \in G(M, R) \text{ s.t. depth}(m) = \text{max depth and } m = b \in B_{enf}$$

A practical reason why one would want to enforce a starting-material constraint is that they may be inexpensive reagents. As such, they can be obtained in larger quantities for use in multi-step synthesis, which necessarily loses material at every synthetic step.

Case 2: Intermediate Constrained Synthesis. A synthesis graph is considered *intermediate constrained* (general case of starting-material constrained) if at least one intermediate node, $m \in G(M, R)$, belongs to B_{enf}:

$$\exists m \in G(M, R) \text{ s.t. } m \in B_{enf}$$

Case 3: Divergent Synthesis. A synthesis graph is considered *divergent* if at least one intermediate node, $m \in G(M, R)$, satisfies both of the following conditions: (1) $m = b \in B_{enf}$, and (2) all $b \in B_{enf}$ are non-commercial. The nuance of *non-commercial* is that they can be highly specific building blocks and potentially much larger in size than common commercial building blocks, which can enable late-stage functionalization (Castellino et al., 2023):

$$\exists m \in G(M, R) \text{ s.t. } \forall b \in B_{enf},\ b \text{ is non-commercial, and } m = b \in B_{enf}$$

TANGO Reward Function. In the context of generative molecular design, previous work has shown that retrosynthesis models can be treated as an oracle and directly optimized for (Guo & Schwaller, 2024c). The effect is that generated molecules are synthesizable, as deemed by retrosynthesis models (from here on, this will just be referred to as "synthesizable", for brevity). In that work, the authors adopted a brute-force approach to *learning* synthesizability, despite the reward signal being binary, i.e., $R \in \{0, 1\}$, denoting whether a molecule is synthesizable or not. This worked because there are enough molecules that are synthesizable, making the optimization landscape not *too* sparse. However, in the constrained synthesis setting, it is highly unlikely that a synthesizable molecule will also contain an enforced building block in its synthesis graph, especially when the number of B_{enf} is small, which is common in real-world applications (Granda et al., 2018; Wołos et al., 2020; 2022). Consequently, this is a very sparse reward environment: without a way to inform the model if it is getting "closer" to achieving the goal, learning becomes extremely challenging. Drawing inspiration from RL, quantifying the degree to which an arbitrary goal is achieved, while intending for another goal is sometimes referred to as *hindsight* (Rauber et al., 2019). Intuitively, defining a reward signal that is not exactly the target objective but is *informative* to achieving the target objective should guide learning (Andrychowicz et al., 2017; Rauber et al., 2019), and can be done through *reward shaping* (Ng et al., 1999; Silver et al., 2021).

In this work, we propose the **TANimoto Group Overlap (TANGO)** reward function that leverages chemistry inductive bias to transform the *sparse* reward environment associated with constrained synthesis, to a *dense* reward environment. Specifically, for every synthesizable molecule, TANGO provides a signal on whether the model is "closer" to incorporating B_{enf}. Given $G(M, R)$, this is achieved by a notion of *similarity* between every node, m, and B_{enf}. We draw inspiration from previous works that leverage Tanimoto similarity (sub-graph similarity) for retrosynthesis (Coley et al., 2017; Zhang et al., 2024). However, Tanimoto similarity alone is insufficient as chemical *reactivity* is often associated with *functional groups* and their neighborhoods which dictate incompatibilities (Molga et al., 2019). Correspondingly, we augment Tanimoto similarity with Functional Group (FG) Overlap (does a given molecule have similar functional groups to B_{enf}?) and Fuzzy Matching Substructure (FMS) (what is the maximum substructure overlap by atom count to B_{enf}?). We note that FMS, if enforcing exact atom hybridization, atom type, chirality, and whether the atom is part of a ring, also *implicitly* considers functional group overlap. In Appendix F, we systematically evaluated various TANGO formulations and their ability to distinguish "goodness" and conclude that equal weighting (0.5) of Tanimoto similarity (TS) and FMS yields the best learnable signal:

$$TANGO(m, B_{enf}) = TS(m, B_{enf}) \cdot 0.5 + FMS(m, B_{enf}) \cdot 0.5 \in [0, 1] \tag{1}$$

We note that since the maximum value of Tanimoto similarity and FMS is 1, TANGO is by design already normalized ∈ [0, 1]. The TANGO reward is the maximum value between all non-root nodes, $m \in G(M, R)$ (Algorithm E). It follows that the *type* of constrained synthesizability can be controlled by a simple toggle of whether only nodes at max depth (*starting-material*) be considered or any node (*intermediate or divergent*).

Molecular Generative Model. Here, we build on Saturn (Guo & Schwaller, 2024b) which is a general-purpose autoregressive language-based model operating on SMILES strings (Weininger, 1988). Saturn uses the Mamba (Gu & Dao, 2023) architecture and performs goal-directed generation using RL. The key mechanism is combining SMILES augmentation (Bjerrum, 2017) with experience replay (Lin, 1992) which directly controls the exploration-exploitation trade-off. In the original work, the authors found that *aggressive* local sampling in chemical space improves sample efficiency across various drug discovery case studies. By contrast, we show that the constrained synthesizability setting necessitates a more exploratory behavior. We pre-train Saturn on PubChem (Kim et al., 2023) after data pre-processing (see Appendix A for details).

Retrosynthesis Model. In this work, we integrate Syntheseus (Maziarz et al., 2023), which is a wrapper around various retrosynthesis models and search algorithms, into Saturn. Through Syntheseus, we use MEGAN (graph-edits based) (Sacha et al., 2021) as the single-step retrosynthesis model coupled with the Retro* (Chen et al., 2020) search algorithm with default hyperparameters. MEGAN was chosen due to its fast inference speed but we emphasize that our framework is retrosynthesis model-agnostic.

Commercial Building Blocks. In this work, B is comprised of the ‘Fragment‘ and ‘Reactive‘ subsets of ZINC (Sterling & Irwin, 2015) (17,721,980) which are part of the commercial building block stock used in the AiZynthFinder (Genheden et al., 2020; Saigiridharan et al., 2024) retrosynthesis model. We note that the size of B is *much* larger than employed in previous synthesizability-constrained works (Gao et al., 2022; Luo et al., 2024; Gao et al., 2024; Seo et al., 2024) (which commonly use Enamine REAL (Grygorenko et al., 2020) US Stock: 223,244 molecules and recently Enamine Comprehensive Catalogue: 1,193,871 molecules). Therefore, an additional result in this paper is showing that our framework can navigate an *enormous* synthesizable space. We also want to highlight that it is straightforward to further increase the size of B, and does not require re-training of the generative model. Next, $B_{enf} \subset B$ is randomly sampled with the following criteria: 150 < molecular weight < 200, no aliphatic carbon chains longer than 3, exclude charged molecules, if rings are present, enforce size $\in \{5, 6\}$, and molecules must contain at least one nitrogen, oxygen, or sulfur atom. We believe this criteria is a reasonable representation of simple building blocks applicable to drug design (see Appendix B for more details) We consider $|B_{enf}| \in \{10, 100\}$ and denote these B_{enf-10} and $B_{enf-100}$, respectively.

Drug Discovery Case Study. The MPO optimization task is to generate molecules with optimized QuickVina2-GPU-2.1 (Trott & Olson, 2010; Alhossary et al., 2015; Tang et al., 2023) docking scores against ATP-dependent Clp protease proteolytic subunit (ClpP) (Mabanglo et al., 2023) (implicated in cancer), high QED (Bickerton et al., 2012), and are synthesizable with either the *starting-material*, *intermediate*, or *divergent synthesis* constraints.

Experimental Details. For method development (see Appendix F for all experimental results), we ran every experiment across 5 seeds (0-4 inclusive) with varying oracle budgets. Once we identified optimal hyperparameters, we ran all main result experiments across 10 seeds (0-9 inclusive) with a 10,000 oracle budget, and reported the wall time to promote practical application. As our framework is, to the best of our knowledge, the first *generative* approach that tackles constrained synthesizability, we focus our investigation on the optimization dynamics and implications of TANGO.

Metrics. We report **Non-solved** and **Solved (Enforced)** as the number of generated molecules that the retrosynthesis model deems unsynthesizable (no route returned) and is synthesizable *with* the presence of an enforced building block, respectively. Note that **Solved (Enforced)** is a much more challenging metric than *just* synthesizable, which previous work has shown is directly learnable (Guo & Schwaller, 2024c). We further report **N** as the number of replicates out of 10 seeds where **Solved (Enforced)** > 0, and the mean and standard deviation for the **# Unique Enforced Blocks**, denoting how many *unique* enforced building blocks are in the routes for the **Solved (Enforced)** molecules. Next, we pool all **Solved (Enforced)** molecules and report the mean and standard deviation of the **# Reaction Steps**. Similarly, we report the mean and standard deviation of docking scores and QED values across varying intervals. Jointly optimizing for constrained synthesizability, minimizing docking scores, and maximizing QED values is the MPO objective and a robust model should be able to achieve this.

4 Results and Discussion

4.1 Making Constrained Synthesizability *Learnable*

Understanding the Optimization Dynamics. In Appendix F, we performed extensive ablation studies to understand the optimization dynamics that enable direct optimization of constrained synthesizability. We summarize our observations (see results in Appendix F.5): firstly, TANGO is the most consistent *learnable* reward function that also enables MPO. While just Tanimoto similarity as the reward function can lead to successful runs, it is less stable (seeds can fail and much higher variance) and MPO is considerably worse, as docking and QED is optimized to a much lesser extent than TANGO. FMS as a reward function is also successful, but generates very few constrained synthesizable molecules. Therefore, through ablation studies, we show that *it takes two to tango* because TANGO's performance is much more consistent and *robust* (across seeds) than using just Tanimoto similarity or FMS alone. Thus far, TANGO was formulated with equal weighting to FMS and Tanimoto similarity. We next investigated varying the weighting, assigning 0.75 to either FMS or Tanimoto Similarity and 0.25 to the other. The results in Appendix F.5 show that putting more weight on Tanimoto similarity leads to more constrained synthesizable molecules, but at the expense of worse MPO. Since MPO is vital for practicality, we chose to designate TANGO with equal weighting the default reward function. Next, we found that the default hyperparameters of Saturn (Guo & Schwaller, 2024b) are *too exploitative* and disadvantageous in this optimization setting. Relaxing this behavior makes constrained synthesizability much more *consistently* learnable (under the fixed oracle budget). Lastly, once molecules with a specific enforced building block are generated, Saturn heavily focuses on that building block, such that within the same generative experiment, often only one *unique* block is enforced (but variable across seeds). We do not consider this a disadvantage as it enables the construction of *divergent synthesis networks* where a common block branches towards many optimal molecules. We discuss the implications of this behavior from a generative perspective in the next section.

Constrained Synthesizability Results. Table 1 shows the results with B_{enf-10} and $B_{enf-100}$ using TANGO (equal weighting). We make the following observations: firstly, all constraints can be learned within the 10,000 oracle budget (approximate 8.5 hours wall time). Secondly, all runs generate non-solvable molecules and many solvable molecules do not contain the enforced blocks, as expected (see **Non-solved** and **Solved (Enforced)**). Nonetheless, generated molecules can achieve docking scores < -10 (considered optimal in previous works and is better than the reference ligand (Koziarski et al., 2024; Guo & Schwaller, 2024c)) and optimal QED values. This demonstrates the capability to perform MPO while also optimizing for constrained synthesizability. Thirdly, **# Unique Enforced Blocks** is relatively low as we observed that once the model incorporates *one* enforced building block, it focuses on generating molecules whose syntheses can be decomposed to that specific block, since the reward it obtains is high and there is a degree of exploitation. Fourthly, the starting-material constraint is more difficult for $B_{enf-100}$ but unexpectedly, not for B_{enf-10}. We speculate the reason for this is exactly due to exploitation behavior. Since TANGO returns the max reward in the synthesis tree (comparing to all B_{enf}), it is possible that more blocks can be a hindrance when there are specific blocks that are particularly favorable. We emphasize that across different seeds, the enforced building blocks can be different, which is important as one could run multiple experiments in parallel and pool the results. Fifthly, we ran the same experiments *without* the QED objective and the optimization task becomes easier (as expected), with higher **Solved (Enforced)** and molecules with docking scores < -10 (Appendix F.6). We ran these sets of experiments for completeness and comparison only, as particularly low QED can result in lipophilic molecules that can be promiscuous binders (Arnott & Planey, 2012). We highlight that the **# Reaction Steps** is generally short, which shows that optimizing for constrained synthesizability does not lead to inefficient synthesis plans. Moreover, amongst the most recent synthesizability-constrained models (Seo et al., 2024), our framework outputs shorter synthesis routes, on average, despite operating in the much more challenging setting. This is because our framework can perform MPO and optimizing for QED implicitly yields shorter synthesis routes, on average, as it constrains the molecular weight. We note that the **# Reaction Steps** in the divergent synthesis results are longer because it takes one step in the first place, to arrive at the divergent blocks. Finally, all runs only took on average, 8.5 hours on a single GPU, which is reasonable, as many commercial drug discovery projects run their generative experiments for 24-72 hours (Livne et al., 2024).

Table 1: Constrained synthesizability results. The reward function is TANGO (equal FMS and Tanimoto similarity weighting). "SM" denotes starting-material constrained. "Unconstrained" denotes the experiment without enforcing building blocks, as a comparison. The mean and standard deviation across 10 seeds (0-9 inclusive) are reported. The number of replicates (out of 10) with at least 1 generated molecule that is synthesizable with an enforced building block is reported with **N**. The number of molecules (pooled across all successful replicates) are partitioned into different docking score thresholds and statistics reported. # Reaction Steps is also reported for the pooled generated molecules that have an enforced block. The total number of molecules in each pool across the 10 seeds is denoted by **M**. For the docking score intervals, we report the scores and QED values. [a] Denotes how many molecule are solvable by the retrosynthesis model. There is no notion of *enforced* in the unconstrained setting.

Configuration	Synthesizability		Docking Score Intervals (QED Annotated)		
	Non-solved	Solved (Enforced)	DS < -10	-10 < DS < -9	-9 < DS < -8
100 Blocks (N=10)	2288 ± 305	2111 ± 1169	-10.36 ± 0.28 (M=487) 0.79 ± 0.09	-9.42 ± 0.24 (M=3096) 0.82 ± 0.09	-8.47 ± 0.26 (M=5904) 0.81 ± 0.09
100 Blocks (SM) (N=10)	1879 ± 186	1524 ± 502	-10.41 ± 0.31 (M=120) 0.78 ± 0.09	-9.41 ± 0.24 (M=985) 0.81 ± 0.09	-8.43 ± 0.25 (M=3156) 0.81 ± 0.09
10 Blocks (N=6)	2425 ± 288	984 ± 1181	-10.38 ± 0.30 (M=659) 0.79 ± 0.10	-9.46 ± 0.25 (M=3981) 0.83 ± 0.09	-8.57 ± 0.25 (M=2419) 0.83 ± 0.10
10 Blocks (SM) (N=9)	2228 ± 182	1004 ± 925	-10.37 ± 0.27 (794) 0.80 ± 0.09	-9.46 ± 0.24 (M=3881) 0.83 ± 0.09	-8.54 ± 0.25 (M=2790) 0.84 ± 0.10
Divergent Blocks (N=4)	2166 ± 202	651 ± 1238	-10.36 ± 0.26 (M=187) 0.84 ± 0.10	-9.41 ± 0.24 (M=1311) 0.86 ± 0.07	-8.48 ± 0.25 (M=2694) 0.86 ± 0.07

Configuration	Synthesizability		Docking Score Intervals (QED Annotated)		
	Non-solved	Solved[a]	DS < -10	-10 < DS < -9	-9 < DS < -8
Unconstrained (N=10)	1827 ± 191	8127 ± 196	-10.36 ± 0.28 (M=5489) 0.87 ± 0.07	-9.42 ± 0.24 (M=20099) 0.88 ± 0.07	-8.47 ± 0.26 (M=26710) 0.87 ± 0.08

Configuration	# Reaction Steps	# Unique Enforced Blocks	Oracle Budget (Wall Time)
100 Blocks	2.37 ± 1.27 (M=21115)	2 ± 0.63	10,000 (8h 31m ± 40m)
100 Blocks (SM)	1.49 ± 0.91 (M=15247)	1.9 ± 0.7	10,000 (8h 33m ± 30m)
10 Blocks	2.70 ± 1.20 (M=9845)	1 ± 0	10,000 (8h 29m ± 30m)
10 Blocks (SM)	2.59 ± 1.04 (M=10040)	1 ± 0	10,000 (8h 39m ± 24m)
Divergent Blocks	3.68 ± 1.08 (M=6512)	1.75 ± 0.83	10,000 (8h 52m ± 42m)

Configuration	# Reaction Steps	# Unique Enforced Blocks	Oracle Budget (Wall Time)
Unconstrained	1.86 ± 1.19 (M=81829)	N/A	10,000 (5h 34m ± 39m)

Are the Results by Chance? While the results thus far were promising, we noticed that many runs (across seeds) converged to the same three enforced building blocks (Fig. 3). We questioned whether the success was simply due to these "lucky" blocks. Therefore, we performed a set of ablation experiments by removing these blocks and re-running all configurations in Table 1. The results show that the model can generate optimal molecules with other enforced building blocks (Appendix F.7). These runs were much less successful (across seeds) but recovered when removing the QED objective. This suggests that the model learns to use certain blocks that are more aligned with the objective function. Next, we further push our framework with an enforced building block set of *5 molecules* dissimilar to the "lucky" blocks. While much more challenging (most runs are unsuccessful under the strict oracle budget), this is still possible, with the routes containing Suzuki coupling reactions (see Appendix F.8 and Fig. 12), which is notably different to the amide coupling reactions in Fig. 3.

Synthesis Networks. Next, we tackle *divergent synthesis* by incorporating larger, non-commercial molecules in the enforced building blocks set. We curated a set of 10 non-commercial blocks from solved synthesis routes (Fig. 3) and ran the same experimental set-up. Table 1 shows that this is also learnable within the oracle budget, albeit much less consistently as only 4/10 seeds were successful. We still argue that our framework is robust as this is a much more challenging task (one can also increase the oracle budget which leads to more successful seeds, as we show in Appendix F.9) and we wanted to show the model can *learn* to enforce these large building blocks *from scratch.* This opens up practical applications for late-stage functionalization, commonly employed in drug discovery (Castellino et al., 2023). Correspondingly, Fig. 3 shows example synthesis networks using results from the *starting-material* and *divergent synthesis* constrained experiments. All generated molecules achieve optimal docking scores (although starting-material constrained resulted in slightly worse

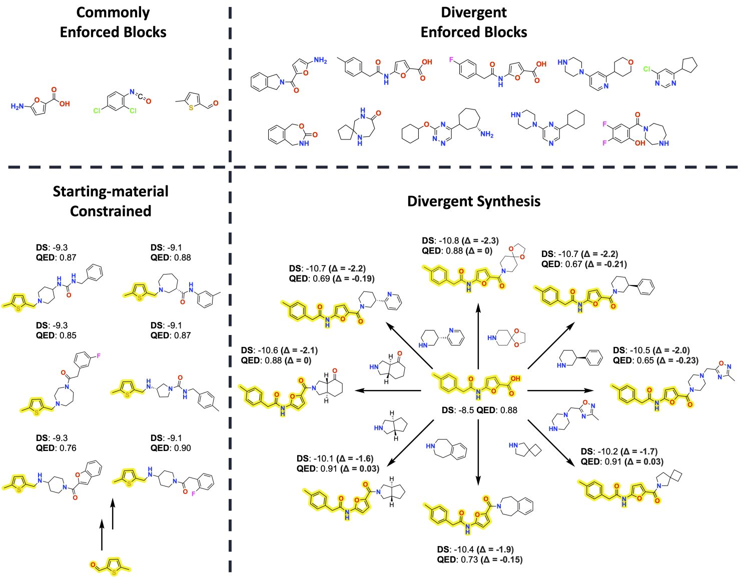


Figure 3: Example generated molecules under the **starting-material** and **divergent synthesis** (one-step synthesis from a non-commercial common intermediate to diverse, high-reward molecules) constraints. The docking scores and QED values are annotated. For the divergent synthesis graph, the Δ docking score (negative is better) and QED (positive is better) are additionally annotated.

scores) and QED values. In the divergent synthesis case, a one-step amide coupling reaction from the enforced block leads to notably improved docking scores, though sometimes with lower QED. Examples of full synthesis routes are shown in Appendix G.

4.2 Learning a Desirable Distribution

Fundamentally, generative models learn to model distributions. In this section, we further demonstrate that TANGO is a *learnable* reward function and that the modeled distribution shifts to satisfy the MPO objective. To do so, we take each final model checkpoint (across the 10 seeds) from the experiment in Table 1 with 100 enforced building blocks (and with QED) and sample 1,000 unique molecules. Fig. 4a shows that a considerable number of sampled molecules are *jointly* synthesizable with an enforced building block (**Solved (Enforced)**). The distribution shift is apparent when compared to 1,000 unique molecules sampled from the *pre-trained model* (before RL), which mostly generates unsynthesizable molecules (**Non-solved**). Fig 4b pools (across the 10 seeds) all the **Solved (Enforced)** molecules and shows the density of docking and QED scores which have shifted towards favorable values. Next, we take the sampled molecules from one seed and plot a UMAP (McInnes et al., 2018) embedding comparing to the molecules sampled from the pre-trained model. It is clear that the checkpoint sampled molecules are dissimilar but we show that the learned distribution is not *perfect*, as the final checkpoint still sometimes samples ill-suited (based on the MPO objective) molecules that are similar to the pre-trained model. Subsequently, we take the sampled molecules from the final model checkpoint and compare the negative log-likelihoods as measured by this checkpoint and the pre-trained model. We make two observations: firstly, the molecules are much more likely under the checkpoint, unsurprisingly. But secondly, and more importantly, the likelihoods from the checkpoint puts more probability mass in a narrower region. We now cross-reference Fig. 4e which shows the top-10 sampled molecules (by docking score) which all share the same enforced building block. The likelihoods are not drastically different, and shows that some exploitation during RL is advantageous as the likelihoods of molecules which share a common structure can be quite similar. Very specifically, given a favorable molecule represented as a SMILES, Saturn's (Guo & Schwaller, 2024b;a) mechanism of optimization involves making it likely to generate *any* SMILES form of the same molecular graph. If it is likely to generate *any* SMILES sequences of the same favorable molecule, small changes to the generated sequence amounts to small chemical changes, which can be advantageous, as similar molecules, on average, have similar properties. The model learns to

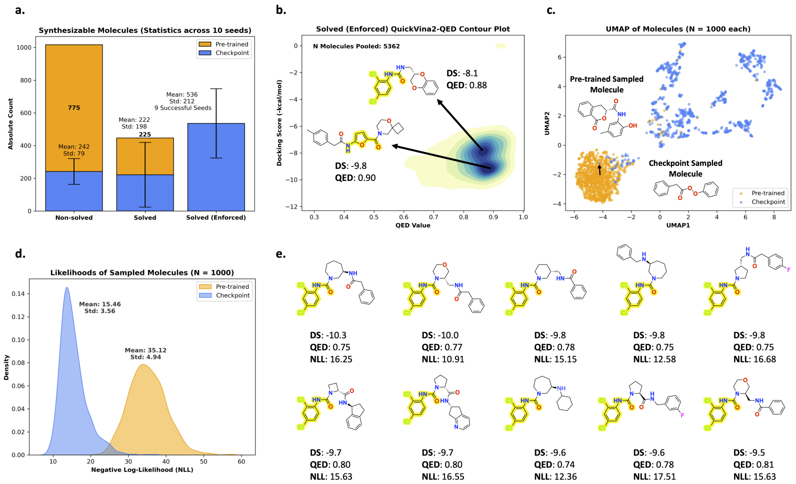


Figure 4: The model learns a distribution of molecules that satisfy the MPO objective. The final model checkpoint from the 100 enforced building blocks experiment (*all 10 seeds*) was used to sample 1,000 *unique* molecules. **a.** Counts of solvable molecules from the checkpoints with the mean and standard deviation reported (non-bolded). 1/10 final model checkpoints was unable to yield "Solved (enforced)" molecules. The pre-trained model (before RL) generates mostly unsynthesizable molecules and no synthesizable molecules with enforced blocks (**metrics are bolded**). **b.** Docking Score (DS) and QED values of the pooled Solved (Enforced) molecules across all seeds. *c, d, e uses 1,000 unique molecules sampled from one final model checkpoint.* **c.** UMAP of sampled molecules compared to the pre-trained model. **d.** Negative log-likelihoods (NLLs) of the sampled molecules. It is much more likely to generate the sampled molecules under the final model checkpoint. **e.** Top-10 (by docking score) molecules with the enforced building block highlighted. The NLLs are similar.

use the building blocks in a way that performs local exploration and assigns a relatively similar likelihood to the neighborhood of molecules. Overall, the results show that taking a *general-purpose* model and *incentivizing* the learning process with TANGO, can shift the modeled distribution to one that captures constrained synthesizability while simultaneously satisfying MPO objectives. This is practically useful, as one could simply sample molecules from model checkpoints to get more desirable molecules (5 seconds to sample 1,000 unique molecules).

5 Conclusion

In this work, we proposed a novel reward function called **TANimoto Group Overlap (TANGO)** that can guide a general-purpose molecular generative model to *directly* optimize for constrained synthesizability while also simultaneously performing MPO. This work is the first example of a *generative* approach for constrained synthesizability, and tackles various degrees of constraints that are practically important in real-world applications: starting-material, intermediate, and divergent synthesis constraints (Fig. 3). The results show that the generative model, Saturn Guo & Schwaller (2024b), when augmented with TANGO, can generate optimal molecules for a drug discovery case study involving molecular docking (Table 1). Moreover, the results show that our framework can learn to enforce building block sets as small as 10 and even 5 (Appendix F.8), which is practically relevant for re-purposing building blocks into useful molecules (Granda et al., 2018; Wołos et al., 2020; 2022). From a generative model perspective, we have shown that optimizing for constrained synthesizability necessitates a better exploration-exploitation trade-off, providing practical insights into MPO in these settings. Furthermore, our results show that *incentivizing* an *unconstrained* model can lead to productive learning even in challenging synthesizability MPO settings. However, "true synthesizability" depends on the accuracy of the retrosynthesis model and they are not perfect. It is likely that some routes generated are not synthetically feasible and/or lack regio- or stereo-selectivity (Molga et al., 2019). This is a limitation of current retrosynthesis models and is an ongoing challenge for improved synthesis planning. As we are not proposing a new retrosynthesis model, this is beyond the scope of this work, but we believe this is important to explicitly acknowledge.

REFERENCES

Amr Alhossary, Stephanus Daniel Handoko, Yuguang Mu, and Chee-Keong Kwoh. Fast, accurate, and reliable molecular docking with quickvina 2. *Bioinformatics*, 31(13):2214–2216, 2015.

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

John A Arnott and Sonia Lobo Planey. The influence of lipophilicity in drug discovery and design. *Expert opinion on drug discovery*, 7(10):863–875, 2012.

G Richard Bickerton, Gaia V Paolini, Jérémy Besnard, Sorel Muresan, and Andrew L Hopkins. Quantifying the chemical beauty of drugs. *Nature chemistry*, 4(2):90–98, 2012.

Esben Jannik Bjerrum. Smiles enumeration as data augmentation for neural network modeling of molecules. *arXiv preprint arXiv:1703.07076*, 2017.

John Bradshaw, Brooks Paige, Matt J Kusner, Marwin Segler, and José Miguel Hernández-Lobato. A model to search for synthesizable molecules. *Advances in Neural Information Processing Systems*, 32, 2019.

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

Nathan J Castellino, Andrew P Montgomery, Jonathan J Danon, and Michael Kassiou. Late-stage functionalization for improving drug-like molecular properties. *Chemical reviews*, 123(13):8127–8153, 2023.

Binghong Chen, Chengtao Li, Hanjun Dai, and Le Song. Retro*: learning retrosynthetic planning with neural guided a* search. In *International conference on machine learning*, pp. 1608–1616. PMLR, 2020.

Shuan Chen and Yousung Jung. Deep retrosynthetic reaction prediction using local reactivity and global attention. *JACS Au*, 1(10):1612–1620, 2021.

Connor W Coley, Luke Rogers, William H Green, and Klavs F Jensen. Computer-assisted retrosynthesis based on molecular similarity. *ACS central science*, 3(12):1237–1245, 2017.

Connor W Coley, Dale A Thomas III, Justin AM Lummiss, Jonathan N Jaworski, Christopher P Breen, Victor Schultz, Travis Hart, Joshua S Fishman, Luke Rogers, Hanyu Gao, et al. A robotic platform for flow synthesis of organic compounds informed by ai planning. *Science*, 365(6453): eaax1566, 2019.

Connor W Coley, Natalie S Eyke, and Klavs F Jensen. Autonomous discovery in the chemical sciences part i: Progress. *Angewandte Chemie International Edition*, 59(51):22858–22893, 2020a.

Connor W Coley, Natalie S Eyke, and Klavs F Jensen. Autonomous discovery in the chemical sciences part ii: outlook. *Angewandte Chemie International Edition*, 59(52):23414–23436, 2020b.

Miruna Cretu, Charles Harris, Julien Roy, Emmanuel Bengio, and Pietro Liò. Synflownet: Towards molecule design with guaranteed synthesis pathways. *arXiv preprint arXiv:2405.01155*, 2024.

Yuanqi Du, Arian R Jamasb, Jeff Guo, Tianfan Fu, Charles Harris, Yingheng Wang, Chenru Duan, Pietro Liò, Philippe Schwaller, and Tom L Blundell. Machine learning-aided generative molecular design. *Nature Machine Intelligence*, pp. 1–16, 2024.

Vendy Fialková, Jiaxi Zhao, Kostas Papadopoulos, Ola Engkvist, Esben Jannik Bjerrum, Thierry Kogej, and Atanas Patronov. Libinvent: reaction-based generative scaffold decoration for in silico library design. *Journal of Chemical Information and Modeling*, 62(9):2046–2063, 2021.

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

Wenhao Gao, Rocío Mercado, and Connor W Coley. Amortized tree generation for bottom-up synthesis planning and synthesizable molecular design. *Proc. 10th International Conference on Learning Representations*, 2022.

Wenhao Gao, Shitong Luo, and Connor W Coley. Generative artificial intelligence for navigating synthesizable chemical space. *arXiv preprint arXiv:2410.03494*, 2024.

Samuel Genheden, Amol Thakkar, Veronika Chadimová, Jean-Louis Reymond, Ola Engkvist, and Esben Bjerrum. Aizynthfinder: a fast, robust and flexible open-source software for retrosynthetic planning. *Journal of cheminformatics*, 12(1):70, 2020.

Gian Marco Ghiandoni, Michael J Bodkin, Beining Chen, Dimitar Hristozov, James EA Wallace, James Webster, and Valerie J Gillet. Renate: a pseudo-retrosynthetic tool for synthetically accessible de novo design. *Molecular Informatics*, 41(4):2100207, 2022.

Gian Marco Ghiandoni, Stuart R Flanagan, Michael J Bodkin, Maria Giulia Nizi, Albert Galera-Prat, Annalaura Brai, Beining Chen, James EA Wallace, Dimitar Hristozov, James Webster, et al. Synthetically accessible de novo design using reaction vectors: Application to parp1 inhibitors. *Molecular Informatics*, 43(4):e202300183, 2024.

Sai Krishna Gottipati, Boris Sattarov, Sufeng Niu, Yashaswi Pathak, Haoran Wei, Shengchao Liu, Simon Blackburn, Karam Thomas, Connor Coley, Jian Tang, et al. Learning to navigate the synthetically accessible chemical space using reinforcement learning. In *International conference on machine learning*, pp. 3668–3679. PMLR, 2020.

Jarosław M Granda, Liva Donina, Vincenza Dragone, De-Liang Long, and Leroy Cronin. Controlling an organic synthesis robot with machine learning to search for new reactivity. *Nature*, 559(7714): 377–381, 2018.

Oleksandr O Grygorenko, Dmytro S Radchenko, Igor Dziuba, Alexander Chuprina, Kateryna E Gubina, and Yurii S Moroz. Generating multibillion chemical space of readily accessible screening compounds. *Iscience*, 23(11), 2020.

Bartosz A Grzybowski, Sara Szymkuć, Ewa P Gajewska, Karol Molga, Piotr Dittwald, Agnieszka Wołos, and Tomasz Klucznik. Chematica: a story of computer code that started to think like a chemist. *Chem*, 4(3):390–398, 2018.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Jeff Guo and Philippe Schwaller. Augmented memory: Sample-efficient generative molecular design with reinforcement learning. *JACS Au*, 2024a.

Jeff Guo and Philippe Schwaller. Saturn: Sample-efficient generative molecular design using memory manipulation. *arXiv preprint arXiv:2405.17066*, 2024b.

Jeff Guo and Philippe Schwaller. Directly optimizing for synthesizability in generative molecular design using retrosynthesis models. *arXiv preprint arXiv:2407.12186*, 2024c.

Yuqiang Han, Xiaoyang Xu, Chang-Yu Hsieh, Keyan Ding, Hongxia Xu, Renjun Xu, Tingjun Hou, Qiang Zhang, and Huajun Chen. Retrosynthesis prediction with an iterative string editing model. *Nature Communications*, 15(1):6404, 2024.

Markus Hartenfeller, Heiko Zettl, Miriam Walter, Matthias Rupp, Felix Reisen, Ewgenij Proschak, Sascha Weggen, Holger Stark, and Gisbert Schneider. Dogs: reaction-driven de novo design of bioactive compounds. *PLoS computational biology*, 8(2):e1002380, 2012.

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

Zygimantas Jocys, Henriette MG Willems, and Katayoun Farrahi. Synthformer: Equivariant pharmacophore-based generation of molecules for ligand-based drug design. *arXiv preprint arXiv:2410.02718*, 2024.

A Peter Johnson, Chris Marshall, and Philip N Judson. Starting material oriented retrosynthetic analysis in the lhasa program. 1. general description. *Journal of chemical information and computer sciences*, 32(5):411–417, 1992.

Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, et al. Pubchem 2023 update. *Nucleic acids research*, 51(D1): D1373–D1380, 2023.

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

Michał Koziarski, Andrei Rekesh, Dmytro Shevchuk, Almer van der Sloot, Piotr Gaiński, Yoshua Bengio, Cheng-Hao Liu, Mike Tyers, and Robert A Batey. Rgfn: Synthesizable molecular generation using gflownets. *arXiv preprint arXiv:2406.08506*, 2024.

Lei Li, Zhuang Chen, Xiwu Zhang, and Yanxing Jia. Divergent strategy in natural product total synthesis. *Chemical reviews*, 118(7):3752–3832, 2018.

Long-Ji Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning*, 8:293–321, 1992.

Bowen Liu, Bharath Ramsundar, Prasad Kawthekar, Jade Shi, Joseph Gomes, Quang Luu Nguyen, Stephen Ho, Jack Sloane, Paul Wender, and Vijay Pande. Retrosynthetic reaction prediction using neural sequence-to-sequence models. *ACS central science*, 3(10):1103–1113, 2017.

Guoqing Liu, Di Xue, Shufang Xie, Yingce Xia, Austin Tripp, Krzysztof Maziarz, Marwin Segler, Tao Qin, Zongzhang Zhang, and Tie-Yan Liu. Retrosynthetic planning with dual value networks. In *International Conference on Machine Learning*, pp. 22266–22276. PMLR, 2023.

Micha Livne, Zulfat Miftahutdinov, Elena Tutubalina, Maksim Kuznetsov, Daniil Polykovskiy, Annika Brundyn, Aastha Jhunjhunwala, Anthony Costa, Alex Aliper, Alán Aspuru-Guzik, et al. nach0: Multimodal natural and chemical languages foundation model. *Chemical Science*, 15(22): 8380–8389, 2024.

Shitong Luo, Wenhao Gao, Zuofan Wu, Jian Peng, Connor W Coley, and Jianzhu Ma. Projecting molecules into synthesizable chemical spaces. *Proc. 41st International Conference on Machine Learning*, 2024.

Mark F Mabanglo, Keith S Wong, Marim M Barghash, Elisa Leung, Stephanie HW Chuang, Afshan Ardalan, Emily M Majaesic, Cassandra J Wong, Shen Zhang, Henk Lang, et al. Potent clpp agonists with anticancer properties bind with improved structural complementarity and alter the mitochondrial n-terminome. *Structure*, 31(2):185–200, 2023.

Krzysztof Maziarz, Austin Tripp, Guoqing Liu, Megan Stanley, Shufang Xie, Piotr Gaiński, Philipp Seidl, and Marwin Segler. Re-evaluating retrosynthesis algorithms with syntheseus. *arXiv preprint arXiv:2310.19796*, 2023.

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Molecule.one. The m1 platform.

Karol Molga, Ewa P Gajewska, Sara Szymkuć, and Bartosz A Grzybowski. The logic of translating chemical knowledge into machine-processable forms: a modern playground for physical-organic chemistry. *Reaction Chemistry & Engineering*, 4(9):1506–1521, 2019.

Rebecca M Neeser, Bruno Correia, and Philippe Schwaller. Fsscore: A machine learning-based synthetic feasibility score leveraging human expertise. *arXiv preprint arXiv:2312.12737*, 2023.

Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, volume 99, pp. 278–287, 1999.

Paulo Rauber, Avinash Ummadisingu, Filipe Mutz, and Jürgen Schmidhuber. Hindsight policy gradients. In *ICLR*, 2019.

Mikołaj Sacha, Mikołaj Błaz, Piotr Byrski, Paweł Dabrowski-Tumanski, Mikołaj Chrominski, Rafał Loska, Paweł Włodarczyk-Pruszynski, and Stanisław Jastrzebski. Molecule edit graph attention network: modeling chemical reactions as sequences of graph edits. *Journal of Chemical Information and Modeling*, 61(7):3273–3284, 2021.

Lakshidaa Saigiridharan, Alan Kai Hassen, Helen Lai, Paula Torren-Peraire, Ola Engkvist, and Samuel Genheden. Aizynthfinder 4.0: developments based on learnings from 3 years of industrial application. *Journal of Cheminformatics*, 16(1):57, 2024.

Philippe Schwaller, Riccardo Petraglia, Valerio Zullo, Vishnu H Nair, Rico Andreas Haeuselmann, Riccardo Pisoni, Costas Bekas, Anna Iuliano, and Teodoro Laino. Predicting retrosynthetic pathways using transformer-based models and a hyper-graph exploration strategy. *Chemical science*, 11(12):3316–3325, 2020.

Marwin HS Segler and Mark P Waller. Neural-symbolic machine learning for retrosynthesis and reaction prediction. *Chemistry–A European Journal*, 23(25):5966–5971, 2017.

Marwin HS Segler, Mike Preuss, and Mark P Waller. Planning chemical syntheses with deep neural networks and symbolic ai. *Nature*, 555(7698):604–610, 2018.

Seonghwan Seo, Jaechang Lim, and Woo Youn Kim. Molecular generative model via retrosynthetically prepared chemical building block assembly. *Advanced Science*, 10(8):2206674, 2023.

Seonghwan Seo, Minsu Kim, Tony Shen, Martin Ester, Jinkyoo Park, Sungsoo Ahn, and Woo Youn Kim. Generative flows on synthetic pathway for drug design. *arXiv preprint arXiv:2410.04542*, 2024.

Jason D Shields, Rachel Howells, Gillian Lamont, Yin Leilei, Andrew Madin, Christopher E Reimann, Hadi Rezaei, Tristan Reuillon, Bryony Smith, Clare Thomson, et al. Aizynth impact on medicinal chemistry practice at astrazeneca. *RSC Medicinal Chemistry*, 15(4):1085–1095, 2024.

David Silver, Satinder Singh, Doina Precup, and Richard S Sutton. Reward is enough. *Artificial Intelligence*, 299:103535, 2021.

Joshua W Sin, Siu Lun Chau, Ryan P Burwood, Kurt Püntener, Raphael Bigler, and Philippe Schwaller. Highly parallel optimisation of nickel-catalysed suzuki reactions through automation and machine intelligence. *ChemRxiv*, 2024.

Megan Stanley and Marwin Segler. Fake it until you make it? generative de novo design and virtual screening of synthesizable molecules. *Current Opinion in Structural Biology*, 82:102658, 2023.

Teague Sterling and John J Irwin. Zinc 15–ligand discovery for everyone. *Journal of chemical information and modeling*, 55(11):2324–2337, 2015.

Felix Strieth-Kalthoff, Han Hao, Vandana Rathore, Joshua Derasp, Théophile Gaudin, Nicholas H Angello, Martin Seifrid, Ekaterina Trushina, Mason Guy, Junliang Liu, et al. Delocalized, asynchronous, closed-loop discovery of organic laser emitters. *Science*, 384(6697):eadk9227, 2024.

Sara Szymkuć, Ewa P Gajewska, Tomasz Klucznik, Karol Molga, Piotr Dittwald, Michał Startek, Michał Bajczyk, and Bartosz A Grzybowski. Computer-assisted synthetic planning: the end of the beginning. *Angewandte Chemie International Edition*, 55(20):5904–5937, 2016.

Shidi Tang, Ji Ding, Xiangyu Zhu, Zheng Wang, Haitao Zhao, and Jiansheng Wu. Vina-gpu 2.1: towards further optimizing docking speed and precision of autodock vina and its derivatives. *bioRxiv*, pp. 2023–11, 2023.

Amol Thakkar, Alain C Vaucher, Andrea Byekwaso, Philippe Schwaller, Alessandra Toniato, and Teodoro Laino. Unbiasing retrosynthesis language models with disconnection prompts. *ACS Central Science*, 9(7):1488–1498, 2023.

Gary Tom, Stefan P Schmid, Sterling G Baird, Yang Cao, Kourosh Darvish, Han Hao, Stanley Lo, Sergio Pablo-García, Ella M Rajaonson, Marta Skreta, et al. Self-driving laboratories for chemistry and materials science. *Chemical Reviews*, 2024.

Oleg Trott and Arthur J Olson. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry*, 31(2):455–461, 2010.

Zhengkai Tu, Sourabh J Choure, Mun Hong Fong, Jihye Roh, Itai Levin, Kevin Yu, Joonyoung F Joung, Nathan Morgan, Shih-Cheng Li, Xiaoqi Sun, et al. Askcos: an open source software suite for synthesis planning. *arXiv preprint arXiv:2501.01835*, 2025.

H Maarten Vinkers, Marc R de Jonge, Frederik FD Daeyaert, Jan Heeres, Lucien MH Koymans, Joop H van Lenthe, Paul J Lewi, Henk Timmerman, Koen Van Aken, and Paul AJ Janssen. Synopsis: synthesize and optimize system in silico. *Journal of medicinal chemistry*, 46(13): 2765–2773, 2003.

Dustin J Vollmann, Lea Winand, and Markus Nett. Emerging concepts in the semisynthetic and mutasynthetic production of natural products. *Current Opinion in Biotechnology*, 77:102761, 2022.

Ian A Watson, Jibo Wang, and Christos A Nicolaou. A retrosynthetic analysis algorithm implementation. *Journal of cheminformatics*, 11:1–12, 2019.

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

Agnieszka Wołos, Rafał Roszak, Anna Żądło-Dobrowolska, Wiktor Beker, Barbara Mikulak-Klucznik, Grzegorz Spólnik, Mirosław Dygas, Sara Szymkuć, and Bartosz A Grzybowski. Synthetic connectivity, emergence, and self-regeneration in the network of prebiotic chemistry. *Science*, 369(6511):eaaw1955, 2020.

Agnieszka Wołos, Dominik Koszelewski, Rafał Roszak, Sara Szymkuć, Martyna Moskal, Ryszard Ostaszewski, Brenden T Herrera, Josef M Maier, Gordon Brezicki, Jonathon Samuel, et al. Computer-designed repurposing of chemical wastes into drugs. *Nature*, 604(7907):668–676, 2022.

Shufang Xie, Rui Yan, Junliang Guo, Yingce Xia, Lijun Wu, and Tao Qin. Retrosynthesis prediction with local template retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 5330–5338, 2023.

Kevin Yu, Jihye Roh, Ziang Li, Wenhao Gao, Runzhong Wang, and Connor W Coley. Double-ended synthesis planning with goal-constrained bidirectional search. *arXiv preprint arXiv:2407.06334*, 2024.

Yemin Yu, Ying Wei, Kun Kuang, Zhengxing Huang, Huaxiu Yao, and Fei Wu. Grasp: Navigating retrosynthetic planning with goal-driven policy. *Advances in Neural Information Processing Systems*, 35:10257–10268, 2022.

Anna Żądło-Dobrowolska, Karol Molga, Olga O Kolodiazhna, Sara Szymkuć, Martyna Moskal, Rafał Roszak, and Bartosz A Grzybowski. Computational synthesis design for controlled degradation and revalorization. *Nature Synthesis*, pp. 1–12, 2024.

Yan Zhang, Xiao He, Shuanhu Gao, Aimin Zhou, and Hao Hao. Evolutionary retrosynthetic route planning [research frontier]. *IEEE Computational Intelligence Magazine*, 19(3):58–72, 2024.

Weihe Zhong, Ziduo Yang, and Calvin Yu-Chian Chen. Retrosynthesis prediction using an end-to-end graph generative architecture for molecular graph editing. *Nature Communications*, 14(1):3009, 2023.

A PubChem Pre-processing and Saturn Pre-training

This section contains the full data pre-processing and pre-training pipeline starting from the raw PubChem which was downloaded from `https://ftp.ncbi.nlm.nih.gov/pubchem/Compound/Extras/`. The exact file is "CID-SMILES.gz".

The exact pre-processing steps along with the SMILES remaining after each step are:

1. Raw PubChem - 118,563,810
2. De-duplication - 118,469,904
3. Standardization (charge and isotope handling) based on `https://github.com/MolecularAI/ReinventCommunity/blob/master/notebooks/Data_Preparation.ipynb`. All SMILES that could not be parsed by RDKit were removed - 109,128,315
4. Tokenize all SMILES based on REINVENT's tokenizer: `https://github.com/MolecularAI/reinvent-models/blob/main/reinvent_models/reinvent_core/models/vocabulary.py`
5. Keep SMILES $\leq$ 80 tokens, 150 $\leq$ molecular weight $\leq$ 650, number of heavy atoms $\leq$ 40, number of rings $\leq$ 8, Size of largest ring $\leq$ 8, longest aliphatic carbon chain $\leq$ 4 - 97,667,549
6. Removed SMILES containing the following tokens (due to undesired chemistry, low token frequency, and redundancy): [Br+2], [Br+3], [Br+], [C+], [C-], [CH+], [CH-], [CH2+], [CH2-], [CH2], [CH], [C], [Cl+2], [Cl+3], [Cl+], [ClH+2], [ClH2+2], [ClH3+3], [N-], [N@+], [N@@+], [NH+], [NH-], [NH2+], [NH3+], [NH], [N], [O+], [OH+], [OH2+], [O], [S+], [S-], [S@+], [S@@+], [S@@], [S@], [SH+], [SH-], [SH2], [SH4], [SH], [S], [c+], [c-], [cH+], [cH-], [c], [n+], [n-], [nH+], [nH], [o+], [s+], [sH+], [sH-], [sH2], [sH4], [sH], [s] - **88,618,780**

The final vocabulary contained 35 tokens (2 extra tokens were added, indicating <START> and <END>) and carbon stereochemistry tokens were kept. Saturn (Guo & Schwaller, 2024b) uses the Mamba (Gu & Dao, 2023) architecture and we used the default hyperparameters in the code-base. With the vocabulary size of 35, the model has 5,265,408 parameters. Saturn was pre-trained for 5 steps, with each step consisting of a full pass through the dataset. The model was pre-trained on a workstation with an NVIDIA RTX 3090 GPU and AMD Ryzen 9 5900X 12-Core CPU. The pre-training parameters were:

1. Training steps = 5
2. Seed = 0
3. Batch size = 512
4. Learning rate = 0.0001
5. Randomize (Bjerrum, 2017) every batch of SMILES

Relevant metrics of the pre-trained model (final model checkpoint) are:

1. Average negative log-likelihood (NLL) = 30.914
2. Validity (10k) = 98.74%
3. Uniqueness (10k) = 98.73%
4. Wall time = 106 hours (takes a relatively long time, though we only used 1 GPU for training. Pre-training also only needs to be done once.)

B Retrosynthesis Details

This section contains details on the retrosynthesis model, the commercial building blocks, and the enforced building blocks.

B.1 Retrosynthesis Framework

In this work, we use Syntheseus (Maziarz et al., 2023) (benchmark platform and wrapper around retrosynthesis models and search algorithms) to run retrosynthesis. We integrate Syntheseus into Saturn (Guo & Schwaller, 2024b) and run the MEGAN (Sacha et al., 2021) single-step model with the Retro* (Chen et al., 2020) search algorithm. In the Syntheseus work, the authors standardize and benchmark many retrosynthesis models and configurations, reporting the inference time and accuracy (across various metrics). We chose MEGAN because it has the fastest inference time, although the top-k accuracies were lower than other models. We note that top-k single-step accuracy does not necessarily equate to better performance on multi-step retrosynthesis. Faster inference time allowed us to iterate experiments and hypotheses faster and is the main reason we chose MEGAN. Our framework is model-agnostic and any retrosynthesis model could be used in place of MEGAN. All MEGAN hyperparameters were tuned by the Syntheseus authors and we use them as is.

B.2 Commercial Building Blocks

All retrosynthesis models require commercial building blocks, B. In this work, we use the ‘Fragment‘ and ‘Reactive‘ sub-sets of ZINC (Sterling & Irwin, 2015), equating to 17,721,980 building blocks. These sub-sets were obtained from the commercial building block stock used in AiZynthFinder (Genheden et al., 2020; Saigiridharan et al., 2024). Next, we consider two sets of enforced building blocks, $B_{enf-10} \subset B_{enf-100} \subset B$. The enforced building block sets (10 or 100) are sub-sets of B and were randomly sampled following the criteria:

1. 150 < molecular weight < 200
2. No aliphatic carbon chains longer than 3
3. Exclude charged building blocks
4. If rings are present, enforce size $\in$ {5, 6}
5. All building blocks must contain at least one nitrogen, oxygen, or sulfur atom

The criteria we defined are based on enforcing building blocks that are "simple, common, and relevant for drug-like molecules". While there is an inherent bias here, we emphasize that our TANGO framework is general and the set of enforced building blocks can be freely changed. Finally, we want to highlight an important implication when considering the commercial building blocks, B, and the generative model. Due to intentional data pre-processing of PubChem, which was used to pre-train Saturn, the generative model cannot generate all the atom types present in B. The specific atom types are phosphorus and silicon. We removed these atoms due to their seldom presence in "drug-like" molecules (although phosphorus is common in pro-drugs). The effect of this is that *some* commercial building blocks are not relevant, but we did not purge these and used the ZINC sub-sets as is. Similar to the enforced building blocks set, the set of commercial building blocks can also freely be changed. The sets of enforced building blocks are provided in the code-base.

C Compute Details

Every experiment (except pre-training Saturn) was run on a cluster equipped with NVIDIA L40S GPUs. As we used a SLURM queuing system, many jobs could be allocated the same GPU to run simultaneously. This makes the wall time for each individual run slower, but the total time to finish experiments is faster. We report the wall times as is.

D Docking Reward Shaping

Saturn expected every property to be optimized to have a normalized reward $\in$ [0, 1]. TANGO and QED are already by design normalized but QuickVina2-GPU docking needs to be reward shaped. This is done by the shaping function shown here.

Algorithm 1: TANGO Reward Calculation

```
Input:
$G(M, R)$ Synthesis graph of generated molecule
$B_{enf}$ Enforced building blocks
max depth Graph depth of terminal leaf nodes
enforce start Boolean flag for starting-material constraint
Output: $reward$

$reward \leftarrow 0$
// Traverse all non-root nodes in the synthesis graph
foreach node $m \in G(M, R)$ and $depth(m) > 0$ do
    // Starting-material constrained or not
    if enforce start then
        if $depth(m) \neq$ max depth then
            continue
        end
    end
    $reward \leftarrow max(reward, \texttt{TANGONodeReward}(m, B_{enf}))$
end
Function TANGONodeReward($node$, $B_{enf}$):
    $node\_reward \leftarrow 0$
    // Loop through all enforced building blocks
    foreach $b_{enf} \in B_{enf}$ do
        // Compute current block's reward
        $TanSim \leftarrow \text{ComputeTanimotoSimilarity}(node, b_{enf})$
        $FMS \leftarrow \text{ComputeFMS}(node, b_{enf})$
        $block\_reward \leftarrow TanSim \cdot 0.50 + FMS \cdot 0.50$
        $node\_reward \leftarrow max(node\_reward, block\_reward)$
    end
    return node_reward
return TANGO_reward
```

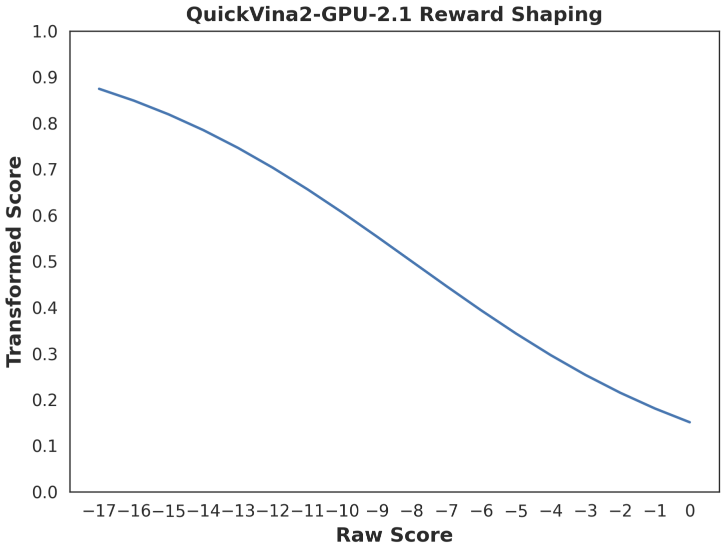


Figure 5: Reward shaping function for docking.

E TANGO Pseudo-code

F TANGO Development and Ablations

In this section, we present the systematic development of TANGO, all ablation studies, and additional results. The section will be divided sequentially into sub-sections detailing our hypotheses, the experiments we ran to study them, and the observations we made. All development experiments were run across 5 seeds (0-4 inclusive) while main result experiments were run across 10 seeds (0-9 inclusive). This information will be noted. The MPO objective is:

1. **Minimize QuickVina2-GPU-2.1** (Trott & Olson, 2010; Alhossary et al., 2015; Tang et al., 2023) docking scores against ATP-dependent Clp protease proteolytic subunit (ClpP) (Mabanglo et al., 2023) (implicated in cancer)
2. **Maximize QED** (Bickerton et al., 2012)
3. **Constrained Synthesizable**, as deemed by the MEGAN (Sacha et al., 2021) retrosynthesis model coupled with Retro* search (Chen et al., 2020)

Next, throughout TANGO development, we change the hyperparameters of Saturn, which directly control for the exploration-exploitation trade-off. We briefly summarize the key hyperparameters and their effect:

1. **Batch Size:** Lower is more exploitative
2. **Augmentation Rounds:** Higher is more exploitative

Finally, for all sets of experiments, we report metrics averaged across either 5 (0-4 inclusive) or 10 (0-9 inclusive) seeds. The number of seeds will be explicitly noted. The metrics are:

1. **Non-solved:** Number of generated molecules that do not have a solved synthetic route
2. **Solved:** Number of generated molecules that have a synthetic route **with at least 1 enforced building block**
3. **Docking Scores - QED:** Average and standard deviation of docking scores and QED values across various docking score thresholds. The rationale for this is because we want to optimize *all* objectives and analyzing different partitions is more informative
4. **Oracle Budget:** Number of oracle calls permitted

5. **Wall Time:** Compute time for the run

Constrained Synthesizability denotes either *start-material constrained* (enforced building blocks appearing at the max depth nodes in the synthesis graph), *intermediate-constrained* (enforced building blocks appearing anywhere in the synthesis graph), or *divergent synthesis* (enforced **non-commercial** building blocks appearing anywhere in the synthesis graph). This information will be explicitly noted. Finally, for brevity, we will write "synthesizable" to mean synthesizable, as deemed by the MEGAN retrosynthesis model.

F.1 How can Constrained Synthesizability be made Learnable?

The starting point of TANGO development drew inspiration from (Coley et al., 2017; Zhang et al., 2024) which used Tanimoto similarity for retrosynthesis problems. We hypothesized that Tanimoto similarity alone is insufficient to inform chemical *reactivity*. Therefore, very initial experiments tried to "filter" nodes by matching for functional groups. Specifically, for every molecule generated that was synthesizable, there is a corresponding synthesis graph whose nodes are every intermediate molecule. The very first reward function traverses these nodes and computes the max Tanimoto similarity to the set of enforced building blocks, *provided that the node overlaps 75% of the functional groups with at least one of the enforced building blocks*, and returns this as the reward. With this initial reward formulation, we used Saturn's (Guo & Schwaller, 2024b) default hyperparameters of batch size 16 and 10 augmentation rounds. These parameters make the model perform local sampling in chemical space *aggressively*. We had run this experiment across 5 seeds (0-4 inclusive) with an oracle budget of 3,000 and only one seed was successful in generating *some* synthesizable molecules with the enforced building blocks. All seeds showed some *learning*, in that the average Tanimoto similarity of the synthesis graphs to the enforced building blocks was increasing (though it always stagnated). At the time, this was highly *irreproducible*, considering only 1/5 runs were successful. However, these failed runs gave us sets of molecules possessing various Tanimoto similarity to the enforced building blocks which we used to investigate various reward shaping functions. Specifically, we took the set of all generated molecules from one of the seeds and partitioned all the molecules that were synthesizable into the following Tanimoto similarity thresholds (to the enforced building blocks):

1. **Low:** 0.0 < TanSim < 0.2 (N = 237)
2. **Med:** 0.2 <= TanSim < 0.3 (N = 438)
3. **Med-High:** 0.3 <= TanSim < 0.4 (N = 734)
4. **High:** 0.4 <= TanSim < 0.5 (N = 38)
5. **Very-High:** 0.5 <= TanSim > 1.0 (N = 712)

The **reward distributions** of these sets of molecules were visualized under different **reward formulations** (Fig. 6). All comparison are to the set of enforced building blocks:

1. **Functional Groups (FG):** Mean or max functional groups overlap
2. **Tanimoto Similarity (TanSim):** Mean or max Tanimoto Similarity
3. **Fuzzy Matching Substructure (FMS):** Mean or max fraction of atoms in the maximum matching substructure
4. **TANGO-FG:** Max TanSim + Mean FG
5. **TANGO-FMS:** Max TanSim + Max FMS
6. **TANGO-All:** Max TanSim + Mean FG + Max FMS

Based on Fig 6, **Max TanSim**, **Max FMS**, and **TANGO-FMS** are able to separate the partitioned Tanimoto similarity intervals the best. These reward formulations are promising because they can distinguish between "closeness" to incorporating the enforced building blocks and enables a gradient for learning. It is important to know that this analysis has an explicit bias: we are assuming that Tanimoto similarity does in fact equate to being "closer", since we partitioned the generated set based on this. However, this gave us the first hypotheses to work with.

Hypotheses:

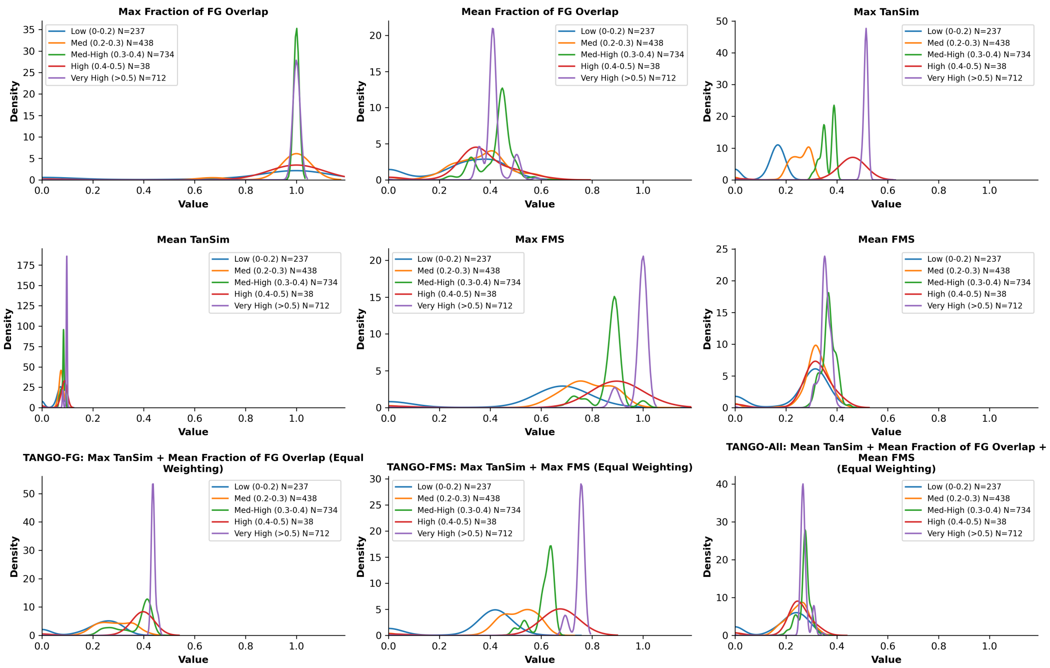


Figure 6: Reward distributions of different reward function formulations.

1. The initial run with 1/5 successful seeds used Batch Size = 16 and Augmentation Rounds = 10. This is likely too *exploitative*. Try a more *exploratory* sampling behavior with **Batch Size = 32 and Augmentation Rounds = 5**.
2. Try the most promising reward functions: **Max TanSim**, **Max FMS**, and **TANGO-FMS**.

Fixed Parameters:

1. **Oracle Budget** = 3,000
2. **Batch Size** = 32
3. **Augmentation Rounds** = 5
4. **Enforced Building Blocks** = 100

Observations: Table 2 shows the results with the mean and standard deviation across 5 seeds (0-4 inclusive). We make the following observations:

1. All reward functions can yield successful runs.
2. FMS and TanSim are inconsistent with 3/5 runs unsuccessful.
3. FMS finds very few molecules satisfying constrained synthesizability
4. TANGO-FMS yields the best average performance.

F.2 Fuzzy Matching Substructure is an Asymmetric Reward Function

The FMS results from the previous section yielded *false positives*: A maximum reward (1.0) was assigned to many generated molecules, yet these molecules did not contain any of the enforced building blocks in its synthesis graphs. The reason for this is due to the asymmetric nature of the designed FMS reward function. We refer to Fig. 7. The FMS reward function computes the maximum substructure overlap and then divides the number of atoms in this overlap by the number of atoms in the enforced building block. Fig. 7 illustrates an edge case where the intermediate node contains

Table 2: Results for Section 1: How can Constrained Synthesizability be made Learnable? The mean and standard deviation across 5 seeds (0-4 inclusive) are reported. The number of replicates (out of 5) with at least 1 generated molecule that is synthesizable with an enforced building block is reported with **N**. The number of molecules (pooled across all successful replicates) are partitioned into different docking score thresholds and statistics reported. # Reaction Steps is also reported for the pooled generated molecules that have an enforced block. The total number of molecules in each pool across the 5 seeds is denoted by **M**. For the docking score intervals, we report the scores and QED values.

Configuration	**Synthesizability**		**Docking Score Intervals (QED Annotated)**		
	Non-solved	**Solved (Enforced)**	**DS < -10**	**-10 < DS < -9**	**-9 < DS < -8**
TanSim (N=2)	577 ± 32	333 ± 408	None N/A	-9.35 ± 0.24 (M=20) 0.70 ± 0.08	-8.38 ± 0.25 (M=248) 0.73 ± 0.10
FMS (N=2)	578 ± 30	9 ± 12	None N/A	None N/A	-8.36 ± 0.26 (M=20) 0.85 ± 0.04
TANGO-FMS (N=5)	643 ± 23	476 ± 377	-10.36 ± 0.25 (M=10) 0.70 ± 0.03	-9.40 ± 0.24 (M=146) 0.70 ± 0.09	-8.42 ± 0.25 (M=596) 0.77 ± 0.10

Configuration	**# Reaction Steps**	**# Unique Enforced Blocks**	**Oracle Budget (Wall Time)**
TanSim	2.53 ± 1.3 (M=1665)	2 ± 0	3,000 (5h 3m ± 24m)
FMS	2 ± 1.07 (M=48)	1.5 ± 0.5	3,000 (4h 29m ± 17m)
TANGO-FMS	2.27 ± 1.28 (M=2382)	1.2 ± 0.4	3,000 (5h 28m ± 45m)

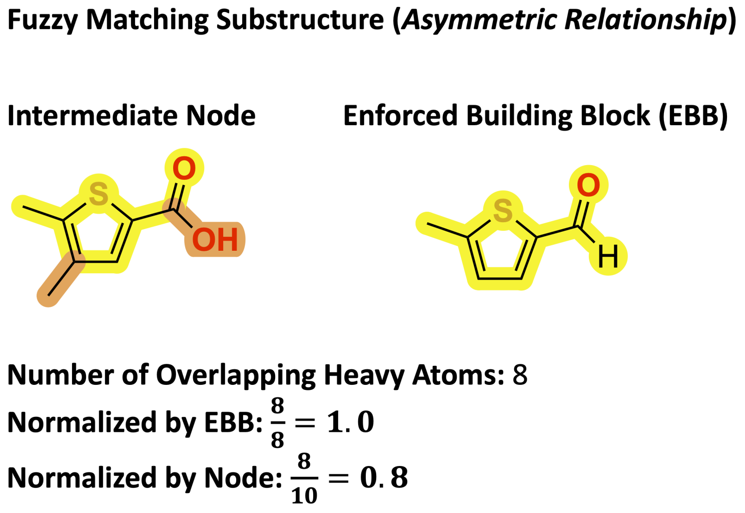

Fuzzy Matching Substructure (*Asymmetric Relationship*)
Intermediate Node
Enforced Building Block (EBB)
S
O
OH
S
O
H
Number of Overlapping Heavy Atoms: 8
Normalized by EBB: $\frac{8}{8} = 1.0$
Normalized by Node: $\frac{8}{10} = 0.8$

Figure 7: Fuzzy Matching Substructure (FMS) is asymmetric depending on whether the number of matching atoms is divided by the number of atoms in the enforced building block or the intermediate node.

the enforced building block as a substructure, but the overall structures do not exactly match. The result was that FMS assigned a perfect reward (1.0). This edge case can be handled by an additional check for exact match, and returning the asymmetric FMS otherwise. This is one possible solution to avoid false positives, yet still reward the model since the overall node and enforced building block structures are similar. **Therefore, for all FMS reward function results, we used this formulation.**

However, we note that false positives only occur in the FMS reward function case, as TANGO-FMS *cannot* yield perfect reward. Since TANGO-FMS is comprised of both FMS and Tanimoto similarity: even if FMS is a false positive, Tanimoto similarity cannot equal 1.0, and thus TANGO-FMS cannot equal 1.0. We hypothesized that this false positive can actually be beneficial, as a perfect reward biases the model towards generating molecules that yield a synthesis graph with an intermediate node *similar* to an enforced building block. This exploitation behavior could be advantageous. In the next section, we investigate the exploration-exploitation trade-off of the generative model when using TANGO-FMS as the reward function. Once we identified optimal hyperparameters, we performed an ablation study in the section after to quantitatively study this asymmetric FMS behavior. We sought to answer whether it is *actually* advantageous to return a "perfect reward (1.0)" for the FMS component in these situations?

F.3 Can we Circumvent Reward Stagnation?

The results from the first section identified TANGO-FMS as the most stable reward function. However, during RL, we observed that the reward improvement often stagnates.

Hypotheses:

1. Further relax the local sampling behavior of Saturn which may help reward stagnation

Fixed Parameters:

1. **Oracle Budget** = 5,000
2. **Batch Size** = 32 or 64
3. **Augmentation Rounds** = Varied
4. **Enforced Building Blocks** = 100

Observations: Table 3 shows the results with the mean and standard deviation across 5 seeds (0-4 inclusive). We make the following observations:

1. Batch32, AR5 is the most successful but imposes a *much longer* wall time. This is due to Saturn's local sampling behavior at low batch sizes and high augmentation rounds.
2. Batch64, AR0 is essentially completely unsuccessful. This affirms that some degree of exploitation is beneficial.
3. Batch64, AR10 is somewhat inconsistent, suggesting *too much* exploitation.
4. Batch64, AR2 and AR5 performs well with the latter notably better, suggesting AR5 may be a good balance between exploration-exploitation.

F.4 Re-visiting Reward Function Formulation for Ablation Studies

The results from the previous section identified tentative hyperparameters with agood balance between exploration-exploitation. With this "better" sampling behavior, we wanted to re-visit the reward function formulations as an extensive ablation to affirm that TANGO-FMS is the best formulation.

Hypotheses:

1. TANGO-FMS may not be the best reward function formulation now that better exploration-exploitation parameters have been identified. Try all reward function formulations.
2. In the previous section, Batch64, AR5 worked much better than Batch64, AR2, but it *might* be too exploitative when we consider moving to a smaller set of enforced blocks and/or starting-material constraints.
3. More thoroughly study the effect of the sampling behavior by increasing the oracle budget.

Fixed Parameters:

1. **Oracle Budget** = 10,000
2. **Batch Size** = 64
3. **Augmentation Rounds** = 2 or 5
4. **Enforced Building Blocks** = 100

Table 3: Results for Section 2: Can we Circumvent Reward Stagnation? The mean and standard deviation across 5 seeds (0-4 inclusive) are reported. The number of replicates (out of 5) with at least 1 generated molecule that is synthesizable with an enforced building block is reported with **N**. Batch denotes "Batch Size" and AR denotes "Augmentation Rounds". The number of molecules (pooled across all successful replicates) is partitioned into different docking score thresholds and statistics are reported. # Reaction Steps is also reported for the pooled generated molecules that have an enforced block. The total number of molecules in each pool across the 5 seeds is denoted by **M**. For the docking score intervals, we report the scores and QED values.

Configuration	**Synthesizability**		**Docking Score Intervals (QED Annotated)**		
	Non-solved	**Solved (Enforced)**	**DS < -10**	**-10 < DS < -9**	**-9 < DS < -8**
Batch32, AR5 (N=4)	1125 ± 110	960 ± 871	-10.34 ± 0.23 (M=11) 0.82 ± 0.08	-9.41 ± 0.25 (M=220) 0.79 ± 0.11	-8.41 ± 0.25 (1093) 0.80 ± 0.09
Batch64, AR10 (N=3)	954 ± 90	674 ± 706	-10.43 ± 0.38 (M=78) 0.68 ± 0.07	-9.46 ± 0.25 (M=208) 0.71 ± 0.10	-8.41 ± 0.25 (484) 0.81 ± 0.09
Batch64, AR5 (N=4)	1029 ± 78	857 ± 529	-10.3 ± 0.25 (M=14) 0.73 ± 0.11	-9.37 ± 0.21 (M=274) 0.78 ± 0.10	-8.44 ± 0.25 (M=1210) 0.81 ± 0.09
Batch64, AR2 (N=4)	1175 ± 89	33 ± 47	-10.7 ± 0 (M=1) 0.75 ± 0	-9.21 ± 0.10 (M=8) 0.82 ± 0.15	-8.39 ± 0.25 (M=49) 0.77 ± 0.11
Batch64, AR0 (N=3)	1921 ± 72	0.60 ± 0.49	None N/A	None N/A	-8.70 ± 0 (M=1) 0.55 ± 0

Configuration	**# Reaction Steps**	**# Unique Enforced Blocks**	**Oracle Budget (Wall Time)**
Batch32, AR5	2.76 ± 1.41 (M=4800)	1 ± 0	5,000 (11h 44m ± 1h 38m)
Batch64, AR10	2.76 ± 1.12 (M=3370)	2 ± 0.82	5,000 (5h 56m ± 25m)
Batch64, AR5	2.13 ± 1.21 (M=4286)	1.25 ± 0.43	5,000 (4h 21m ± 18m)
Batch64, AR2	1.60 ± 0.82 (M=1234)	2 ± 1	5,000 (3h 20m ± 7m)
Batch64, AR0	1 ± 0 (M=3)	1 ± 0	5,000 (2h 52m ± 2m)

Observations: Table 4 shows the results with the mean and standard deviation across 5 seeds (0-4 inclusive). We make the following observations:

1. Surprisingly, Brute-force is sometimes successful but is inconsistent, as expected. Notably many molecules are non-solved (no retrosynthesis route found).
2. FG poorly distinguishes between "goodness" and is essentially unsuccessful, as expected.
3. FMS can distinguish between "goodness" but is not very successful, somewhat unexpectedly.
4. TamSim continues to be successful but is inconsistent, mirroring initial results.
5. TANGO with components of FG are more unsuccessful, in agreement with FG being a poor reward function formulation.
6. TANGO-FMS is most stable, mirroring initial results.
7. TANGO-FMS with the "Asymmetric FMS" implementation (Appendix F.2) performs worse than without. The variance for Solved (Enforced) is higher and much fewer molecules with good docking scores and QED are generated. For this reason, from here on, the original FMS implementation is used, as detailed in Appendix F.2.
8. TANGO-FMS (but with AR5) can outperform TANGO-FMS (AR2) but is notably more inconsistent. This affirms our hypothesis that AR5 might be too exploitative. Importantly, the runs with AR5 also have a much longer wall time, again, due to Saturn's local sampling behavior. Based on these results, AR2 is likely a better balance between exploration-exploitation.

F.5 In TANGO-FMS, is either FMS or Tanimoto Similarity more Important?

The results from the previous section identified hyperparameters with good balance between exploration-exploitation. Thus far, all TANGO formulations weight each component equally. The next question we asked was whether certain components were more important?

Table 4: Results for Section 3: Re-visiting Reward Function Formulation for Ablation Studies. The mean and standard deviation across 5 seeds (0-4 inclusive) are reported. The number of replicates (out of 5) with at least 1 generated molecule that is synthesizable with an enforced building block is reported with **N**. AR denotes "Augmentation Rounds". The number of molecules (pooled across all successful replicates) is partitioned into different docking score thresholds and statistics are reported. # Reaction Steps is also reported for the pooled generated molecules that have an enforced block. The total number of molecules in each pool across the 5 seeds is denoted by **M**. For the docking score intervals, we report the scores and QED values.

Configuration	**Synthesizability**		**Docking Score Intervals (QED Annotated)**		
	Non-solved	**Solved (Enforced)**	**DS < -10**	**-10 < DS < -9**	**-9 < DS < -8**
Brute-force (N=3)	5547 ± 1554	2175 ± 1954	-10.36 ± 0.25 (M=25) 0.46 ± 0.16	-9.33 ± 0.22 (M=529) 0.60 ± 0.19	-8.43 ± 0.25 (M=3196) 0.69 ± 0.19
TanSim (N=3)	2275 ± 143	1863 ± 1827	-10.36 ± 0.24 (M=30) 0.72 ± 0.09	-9.36 ± 0.23 (M=487) 0.76 ± 0.10	-8.43 ± 0.25 (M=2389) 0.79 ± 0.09
FG (N=4)	2144 ± 263	1 ± 1	None N/A	-9.20 ± 0 (M=1) 0.87 ± 0	-8.50 ± 0.28 (M=2) 0.85 ± 0.03
FMS (N=5)	1693 ± 174	114 ± 205	-10.36 ± 0.40 (M=5) 0.75 ± 0.15	-9.41 ± 0.25 (M=53) 0.85 ± 0.09	-8.48 ± 0.25 (M=173) 0.85 ± 0.07
TANGO-FG (N=5)	1957 ± 203	658 ± 967	-10.20 ± 0.10 (M=9) 0.74 ± 0.07	-9.27 ± 0.18 (M=205) 0.78 ± 0.10	-8.48 ± 0.25 (M=1280) 0.82 ± 0.09
TANGO-FMS (N=5)	2229 ± 325	1743 ± 715	-10.34 ± 0.25 (M=218) 0.77 ± 0.11	-9.44 ± 0.25 (M=1606) 0.83 ± 0.10	-8.49 ± 0.26 (M=2206) 0.82 ± 0.10
TANGO-FMS Asymmetric-FMS (N=5)	2249 ± 323	1866 ± 1083	-10.36 ± 0.27 (M=59) 0.71 ± 0.10	-9.39 ± 0.25 (M=513) 0.79 ± 0.10	-8.40 ± 0.25 (M=2049) 0.80 ± 0.09
TANGO-FMS (AR5) (N=4)	2157 ± 182	2521 ± 2060	-10.29 ± 0.18 (M=11) 0.71 ± 0.12	-9.33 ± 0.22 (M=382) 0.80 ± 0.11	-8.40 ± 0.24 (M=2881) 0.83 ± 0.09
TANGO-All (N=3)	2049 ± 93	147 ± 245	-10.43 ± 0.27 (M=31) 0.74 ± 0.08	-9.41 ± 0.26 (M=227) 0.82 ± 0.09	-8.53 ± 0.25 (M=283) 0.84 ± 0.09

Configuration	**# Reaction Steps**	**# Unique Enforced Blocks**	**Oracle Budget (Wall Time)**
Brute-force	3 ± 1.54 (M=10876)	1.33 ± 0.47	10,000 (7h 29m ± 2h 11m)
TanSim	2.18 ± 1.16 (M=9319)	1.33 ± 0.47	10,000 (9h 11m ± 48m)
FG	3.33 ± 1.80 (M=9)	1.75 ± 0.83	10,000 (7h 23m ± 16m)
FMS	1.78 ± 1.04 (M=570)	1.8 ± 0.75	10,000 (7h 50 ± 26)
TANGO-FG	2.21 ± 1.15 (M=3237)	1.8 ± 0.75	10,000 (8h 29m ± 25m)
TANGO-FMS	2.35 ± 1.24 (M=8719)	2.2 ± 0.75	10,000 (8h 12m ± 15m)
TANGO-FMS (Asymmetric-FMS)	2.24 ± 1.19 (M=9334)	2.2 ± 0.4	10,000 (8h 42m ± 26m)
TANGO-FMS (AR5)	2.58 ± 1.17 (M=12608)	1.5 ± 0.5	10,000 (12h 36m ± 52m)
TANGO-All	2.74 ± 1.18 (M=714)	2 ± 0.82	10,000 (8h 11m ± 12m)

Hypotheses:

1. FMS should be more informative than Tanimoto similarity to inform chemical *reactivity*. Test the effect of components weighting.

Fixed Parameters:

1. **Oracle Budget** = 10,000
2. **Batch Size** = 64
3. **Augmentation Rounds** = 2
4. **Enforced Building Blocks** = 100

Observations: Table 5 shows the results with the mean and standard deviation across 5 seeds (0-4 inclusive). "High" indicates 0.75 weighting while the other component is 0.25. TANGO-FMS has equal weighting (0.5 FMS, 0.5 Tanimoto). We make the following observations:

1. TANGO-FMS with equal weighting performs the best in the context of MPO as docking scores are better.

2. TANGO-FMS-High-TanSim generates more solved molecules but docking scores are worse. These suggests suggest that MPO is better with TANGO-FMS (equal weighting) and is the reward function we use from here on.

Table 5: Results for Section 4: In TANGO-FMS, is either FMS or Tanimoto Similarity more Important? The mean and standard deviation across 5 seeds (0-4 inclusive) are reported. The number of replicates (out of 5) with at least 1 generated molecule that is synthesizable with an enforced building block is reported with **N**. The number of molecules (pooled across all successful replicates) is partitioned into different docking score thresholds and statistics are reported. # Reaction Steps is also reported for the pooled generated molecules that have an enforced block. The total number of molecules in each pool is denoted by **M**. For the docking score intervals, we report the scores and QED values.

Configuration	**Synthesizability**		**Docking Score Intervals (QED Annotated)**		
	Non-solved	**Solved (Enforced)**	**DS < -10**	**-10 < DS < -9**	**-9 < DS < -8**
TANGO (0.50 FMS, 0.50 TanSim) (N=5)	2229 ± 325	1743 ± 715	-10.34 ± 0.25 (M=218) 0.77 ± 0.11	-9.44 ± 0.25 (M=1606) 0.83 ± 0.10	-8.49 ± 0.26 (M=2206) 0.82 ± 0.10
TANGO (0.75 FMS, 0.25 TanSim) (N=5)	1962 ± 166	1725 ± 747	-10.30 ± 0.17 (M=23) 0.78 ± 0.06	-9.32 ± 0.22 (M=498) 0.83 ± 0.07	-8.44 ± 0.25 (M=2554) 0.85 ± 0.07
TANGO (0.25 FMS, 0.75 TanSim) (N=5)	2464 ± 437	2737 ± 1038	-10.31 ± 0.24 (M=84) 0.75 ± 0.10	-9.39 ± 0.25 (M=837) 0.78 ± 0.10	8.43 ± 0.25 (M=3468) 0.80 ± 0.10

Configuration	**# Reaction Steps**	**# Unique Enforced Blocks**	**Oracle Budget (Wall Time)**
TANGO (0.5 FMS, 0.5 TanSim)	2.35 ± 1.24 (M=8719)	2.2 ± 0.75	10,000 (8h 12m ± 15m)
TANGO (0.75 FMS, 0.25 TanSim)	2.39 ± 1.24 (M=8625)	1.6 ± 0.49	10,000 (8h 36m ± 16m)
TANGO (0.25 FMS, 0.75 TanSim)	2.30 ± 1.30 (M=13688)	1.6 ± 0.49	10,000 (8h 51m ± 26m)

F.6 Investigating Robustness

With optimal hyperparameters identified, we expand to robustness studies and run every experiment across 10 seeds (0-9 inclusive) and investigate enforcing a smaller set of building blocks. We also probe whether the starting-material constraint is also learnable within the oracle budget. Finally, we also perform a set of experiments *without* the QED objective.

Fixed Parameters:

1. **Oracle Budget** = 10,000
2. **Batch Size** = 64
3. **Augmentation Rounds** = 2
4. **Reward Function** = TANGO-FMS (equal weighting)
5. **Enforced Building Blocks** = 100

Observations: Table 6 shows the results with the mean and standard deviation across 10 seeds (0-9 inclusive). We make the following observations:

1. All constraints are learnable.
2. When not enforcing QED, the model generates many more molecules with "good" docking scores, and expectedly, at the expense of QED. This affirms that the MPO is tunable, allowing tailored design of molecules that are also constrained by synthesis.
3. As expected, when not enforcing QED, the average reaction steps is longer, since QED constrains molecular weight.

F.7 Lucky Building Blocks?

From the previous set of experiments, we noticed that the generative model was always incorporating the same 3 enforced building blocks. One in particular was especially common, such that most runs

Table 6: Results for Section 5: Investigating Robustness. "SM" denotes starting-material constrained. The mean and standard deviation across 10 seeds (0-9 inclusive) are reported. The number of replicates (out of 10) with at least 1 generated molecule that is synthesizable with an enforced building block is reported with **N**. TThe number of molecules (pooled across all successful replicates) is partitioned into different docking score thresholds and statistics are reported. # Reaction Steps is also reported for the pooled generated molecules that have an enforced block. The total number of molecules in each pool across the 10 seeds is denoted by **M**. For the docking score intervals, we report the scores and QED values.

Configuration	**Synthesizability**		**Docking Score Intervals (QED Annotated)**		
	Non-solved	**Solved (Enforced)**	**DS < -10**	**-10 < DS < -9**	**-9 < DS < -8**
100 Blocks (N=10)	2288 ± 305	2111 ± 1169	-10.36 ± 0.28 (M=487) 0.79 ± 0.09	-9.42 ± 0.24 (M=3096) 0.82 ± 0.09	-8.47 ± 0.26 (M=5904) 0.81 ± 0.09
100 Blocks (no QED) (N=10)	1848 ± 158	3723 ± 681	-10.74 ± 0.55 (M=8649) 0.23 ± 0.06	-9.48 ± 0.26 (M=10633) 0.27 ± 0.11	-8.53 ± 0.25 (M=9771) 0.32 ± 0.15
100 Blocks (SM) (N=10)	1879 ± 186	1524 ± 502	-10.41 ± 0.31 (M=120) 0.78 ± 0.09	-9.41 ± 0.24 (M=985) 0.81 ± 0.09	-8.43 ± 0.25 (M=3156) 0.81 ± 0.09
100 Blocks (SM, no QED) (N=10)	1734 ± 172	1189 ± 963	-10.50 ± 0.40 (M=685) 0.31 ± 0.15	-9.43 ± 0.25 (M=2357) 0.38 ± 0.17	-8.49 ± 0.25 (M=4121) 0.45 ± 0.18
10 Blocks (N=6)	2425 ± 288	984 ± 1181	-10.38 ± 0.30 (M=659) 0.79 ± 0.10	-9.46 ± 0.25 (M=3981) 0.83 ± 0.09	-8.57 ± 0.25 (M=2419) 0.83 ± 0.10
10 Blocks (no QED) (N=9)	1967 ± 211	2640 ± 1066	-10.51 ± 0.39 (M=3453) 0.35 ± 0.16	-9.47 ± 0.25 (M=8402) 0.39 ± 0.16	-8.54 ± 0.25 (M=8332) 0.41 ± 0.16
10 Blocks (SM) (N=9)	2228 ± 182	1004 ± 925	-10.37 ± 0.27 (794) 0.80 ± 0.09	-9.46 ± 0.24 (M=3881) 0.83 ± 0.09	-8.54 ± 0.25 (M=2790) 0.84 ± 0.10
10 Blocks (SM, no QED) (N=8)	1753 ± 147	1563 ± 1111	-10.57 ± 0.45 (M=2439) 0.35 ± 0.15	-9.47 ± 0.25 (M=5120) 0.43 ± 0.17	-8.54 ± 0.25 (M=4649) 0.44 ± 0.17

Configuration	**# Reaction Steps**	**# Unique Enforced Blocks**	**Oracle Budget (Wall Time)**
100 Blocks	2.37 ± 1.27 (M=21115)	2 ± 0.63	10,000 (8h 31m ± 40m)
100 Blocks (no QED)	3.24 ± 1.20 (M=37231)	1.8 ± 0.75	10,000 (7h 27m ± 13m)
100 Blocks (SM)	1.49 ± 0.91 (M=15247)	1.9 ± 0.7	10,000 (8h 33m ± 30m)
100 Blocks (SM, no QED)	2.34 ± 1.18 (M=11890)	1.7 ± 0.64	10,000 (8h 3m ± 33m)
10 Blocks	2.70 ± 1.20 (M=9845)	1 ± 0	10,000 (8h 29m ± 30m)
10 Blocks (no QED)	3.18 ± 1.25 (M=26403)	1.22 ± 0.42	10,000 (7h 51m ± 33m)
10 Blocks (SM)	2.59 ± 1.04 (M=10040)	1 ± 0	10,000 (8h 39m ± 24m)
10 Blocks (SM, no QED)	2.65 ± 0.88 (M=15632)	1 ± 0	10,000 (8h 9m ± 27m)

using the set of *10* enforced blocks, use it. We questioned whether TANGO's success was due to luck in having "suitable" building blocks. Therefore, we perform further ablation experiments that purge these 3 building blocks. Similar to the previous set of experiments, we run every configuration here across 10 seeds (0-9 inclusive).

Fixed Parameters:

1. **Oracle Budget** = 10,000
2. **Batch Size** = 64
3. **Augmentation Rounds** = 2
4. **Reward Function** = TANGO-FMS (equal weighting)
5. ***Purged* Enforced Building Blocks** = 97 (Purged the 3 common enforced building blocks from the set of 100)

Observations: Table 7 shows the results with the mean and standard deviation across 10 seeds (0-9 inclusive). We make the following observations:

1. Other building blocks can be enforced.
2. The runs become less consistent (less successful seeds out of 10). Runs without QED are consistently succcessful, suggesting that the *commonly* enforced blocks were chosen due to being able to jointly satisfy QED and docking.

Table 7: Results for Section 6: Lucky Building Blocks? "SM" denotes starting-material constrained. The mean and standard deviation across 10 seeds (0-9 inclusive) are reported. The number of replicates (out of 10) with at least 1 generated molecule that is synthesizable with an enforced building block is reported with **N**. The number of molecules (pooled across all successful replicates) is partitioned into different docking score thresholds and statistics are reported. # Reaction Steps is also reported for the pooled generated molecules that have an enforced block. The total number of molecules in each pool across the 10 seeds is denoted by **M**. For the docking score intervals, we report the scores and QED values.

Configuration	**Synthesizability**		**Docking Score Intervals (QED Annotated)**		
	Non-solved	**Solved (Enforced)**	**DS $<$ -10**	**-10 $<$ DS $<$ -9**	**-9 $<$ DS $<$ -8**
100 Blocks Purged (N=4)	2322 ± 233	77 ± 229	-10.56 ± 0.37 (M=17) 0.85 ± 0.03	-9.38 ± 0.24 (M=117) 0.87 ± 0.05	-8.48 ± 0.25 (M=376) 0.89 ± 0.04
100 Blocks Purged (no QED) (N=9)	1794 ± 193	1553 ± 1211	-10.65 ± 0.49 (M=2649) 0.25 ± 0.11	-9.47 ± 0.26 (M=4345) 0.30 ± 0.15	-8.52 ± 0.25 (M=4648) 0.36 ± 0.17
100 Blocks Purged (SM) (N=5)	2179 ± 298	166 ± 333	-10.30 ± 0.17 (M=6) 0.83 ± 0.08	-9.39 ± 0.25 (M=128) 0.83 ± 0.09	-8.44 ± 0.24 (M=636) 0.87 ± 0.07
100 Blocks Purged (SM, no QED) (N=8)	1688 ± 239	1456 ± 1112	-10.49 ± 0.38 (M=1032) 0.34 ± 0.11	-9.43 ± 0.25 (M=3871) 0.36 ± 0.13	-8.52 ± 0.25 (M=5624) 0.37 ± 0.15

Configuration	**# Reaction Steps**	**# Unique Enforced Blocks**	**Oracle Budget (Wall Time)**
100 Blocks Purged	5.97 ± 1.17 (M=769)	1.25 ± 0.43	10,000 (8h 54m ± 20m)
100 Blocks Purged (no QED)	3.35 ± 1.19 (M=15525)	1.44 ± 0.50	10,000 (7h 15m ± 12m)
100 Blocks Purged (SM)	4.12 ± 2.29 (M=1660)	1.2 ± 0.4	10,000 (8h 41m ± 28m)
100 Blocks Purged (SM, no QED)	3.30 ± 1.28 (M=14562)	1.62 ± 0.70	10,000 (7h 40m ± 25m)

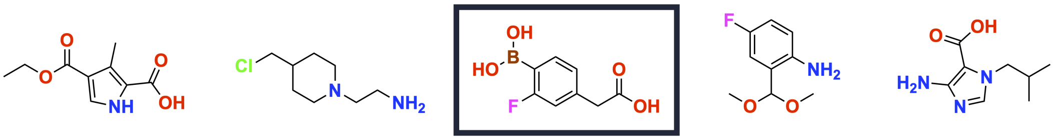


Figure 8: 5 Enforced Building Blocks Set. The circled block is the Suzuki coupling reagent used in all the successful runs without QED (N=2/10 seeds).

F.8 5 Enforced Blocks

We next push our framework further by curating 5 building blocks (Fig. 8) that are dissimilar and/or can be involved in *different* reaction chemistries. Our objective was to investigate whether the model can learn to incorporate such a small set of blocks and whether other chemical reactions can be enforced.

Fixed Parameters:

1. **Oracle Budget** = 10,000 or 15,000
2. **Batch Size** = 64
3. **Augmentation Rounds** = 2
4. **Reward Function** = TANGO-FMS (equal weighting)
5. **Enforced Building Blocks** = 5 (dissimilar to the ones used thus far)

Observations: Table 8 shows the results with the mean and standard deviation across 10 seeds (0-9 inclusive). We make the following observations:

1. The task is challenging under the 10,000 oracle budget when QED is also optimized for.
2. Without optimizing for QED and increasing the oracle budget to 15,000 results in some successes (2/10 seeds).
3. The two successful replicates both enforced only the Suzuki block (Boron containing) which is circled in Fig. 8.

4. The results here show that learning to enforce such a small set of building blocks is possible. In practice, one could further increase the oracle budget which we did not explore due to time limits on the cluster we used. The two successful replicates (with a 15,000 oracle budget) took about 12.5 hours which we believe is still very reasonable.

Table 8: Results for Section 7: 5 Enforced Blocks. The mean and standard deviation across 10 seeds (0-9 inclusive) are reported. The number of replicates (out of 10) with at least 1 generated molecule that is synthesizable with an enforced building block is reported with **N**. The number of molecules (pooled across all successful replicates) is partitioned into different docking score thresholds and statistics are reported. # Reaction Steps is also reported for the pooled generated molecules that have an enforced block. The total number of molecules in each pool is denoted by **M**. For the docking score intervals, we report the scores and QED values.

Configuration	**Synthesizability**		**Docking Score Intervals (QED Annotated)**		
	Non-solved	**Solved (Enforced)**	**DS < -10**	**-10 < DS < -9**	**-9 < DS < -8**
5 Blocks (N=0)	2639 ± 186	0 ± 0	N/A N/A	N/A N/A	N/A N/A
5 Blocks (no QED, 15k Budget) (N=2)	3333 ± 437	972 ± 2112	-11.73 ± 0.93 (M=7044) 0.29 ± 0.07	-9.51 ± 0.26 (M=1419) 0.38 ± 0.13	-8.59 ± 0.24 (M=670) 0.39 ± 0.16

Configuration	**# Reaction Steps**	**# Unique Enforced Blocks**	**Oracle Budget (Wall Time)**
5 Blocks	N/A	N/A	10,000 (9h 24m ± 25m)
5 Blocks (no QED, 15k Budget)	3.79 ± 0.83 (M=9723)	1 ± 0	15,000 (12h 33m ± 34m)

F.9 DIVERGENT SYNTHESIS

Often, divergent synthesis (Li et al., 2018) is desirable, whereby intermediates (usually non-commercially available) are enforced in the synthesis path. This can be used for late-stage functionalization (Castellino et al., 2023) which is particularly relevant in drug discovery to explore SAR. In this section, we select intermediate non-commercial blocks from solved paths. We note that this is artificial in the sense that these selected intermediates were taken from solved routes, and are likely "favorable". However, we were interested in whether a model can learn *from scratch* to enforce relatively large building blocks. For this reason, we curated 10 selected intermediates and investigate the ability of TANGO to learn divergent synthesis constraints.

Fixed Parameters:

1. **Oracle Budget** = 10,000 or 15,000
2. **Batch Size** = 64
3. **Augmentation Rounds** = 2
4. **Reward Function** = TANGO-FMS (equal weighting)
5. ***Divergent* Enforced Building Blocks** = 10 (Curated from successful runs)

Observations: Table 9 shows the results with the mean and standard deviation across 10 seeds (0-9 inclusive). We make the following observations:

1. Divergent blocks can be enforced but the runs are less consistently successful than with the original sets of enforced building blocks, under a 10,000 oracle budget.
2. The runs do not necessarily take longer which means that in practical applications, one could increase the oracle budget. We believe that the wall times of all our experiments (7-9 hours) are reasonable and that much longer is tolerable in real-world applications (<= 24h and even > 24h if the model can truly solve the MPO task).
3. Increasing the oracle budget to 15,000 results in more successful seeds. Therefore, simply using more compute (within reason) is a straightforward solution.

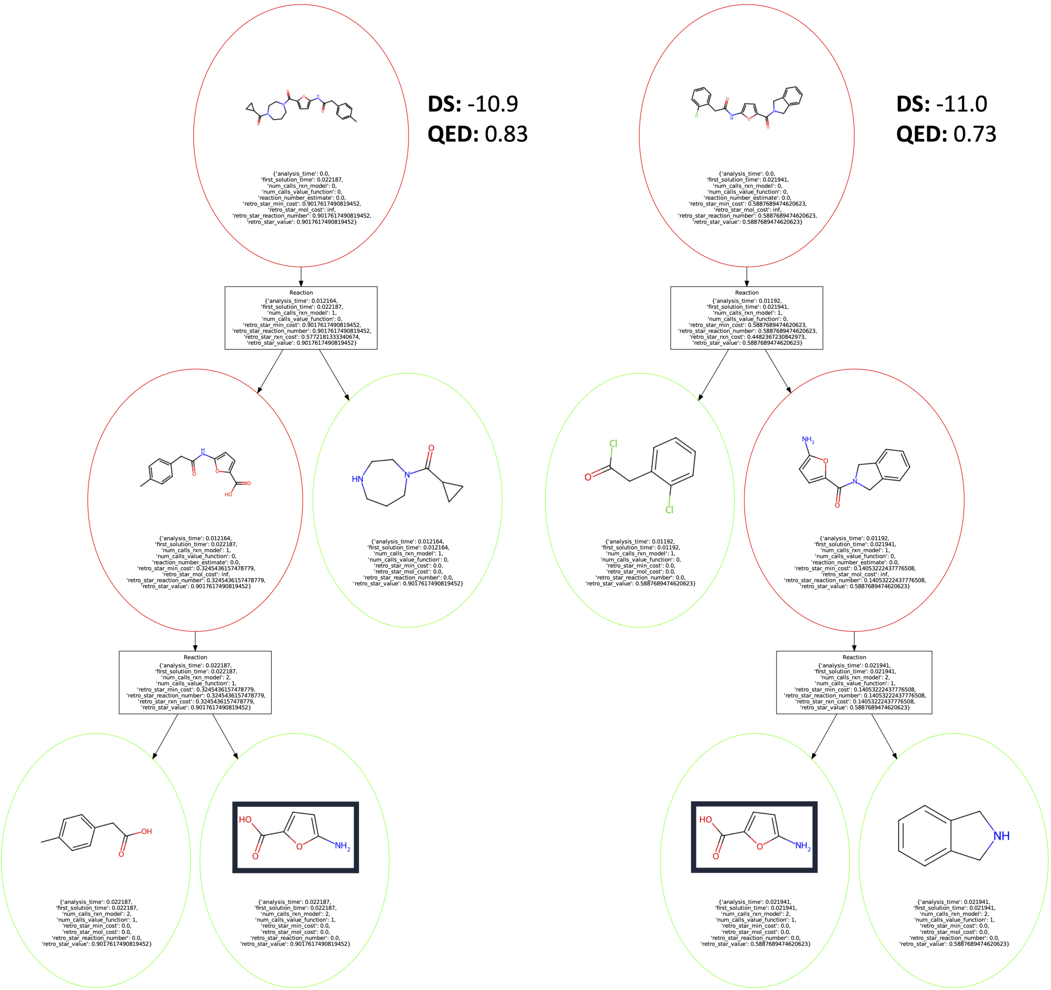
Table 9: Results for Section 8: Divergent Synthesis. The mean and standard deviation across 10 seeds (0-9 inclusive) are reported. The number of replicates (out of 10) with at least 1 generated molecule that is synthesizable with an enforced building block is reported with **N**. The number of molecules (pooled across all successful replicates) is partitioned into different docking score thresholds and statistics are reported. # Reaction Steps is also reported for the pooled generated molecules that have an enforced block. The total number of molecules in each pool across the 10 seeds is denoted by **M**. For the docking score intervals, we report the scores and QED values.

Configuration	**Synthesizability**		**Docking Score Intervals (QED Annotated)**		
	Non-solved	**Solved (Enforced)**	**DS < -10**	**-10 < DS < -9**	**-9 < DS < -8**
Divergent Blocks (N=4)	2166 ± 202	651 ± 1238	-10.36 ± 0.26 (M=187) 0.84 ± 0.10	-9.41 ± 0.24 (M=1311) 0.86 ± 0.07	-8.48 ± 0.25 (M=2694) 0.86 ± 0.07
Divergent Blocks (15k Budget) (N=5)	3720 ± 631	1519 ± 2321	-10.36 ± 0.25 (M=538) 0.82 ± 0.08	-9.44 ± 0.25 (M=3191) 0.85 ± 0.08	-8.47 ± 0.25 (M=6324) 0.87 ± 0.08
Divergent Blocks (no QED) (N=3)	1937 ± 210	540 ± 1259	-10.61 ± 0.47 (M=1099) 0.29 ± 0.11	-9.48 ± 0.25 (M=1894) 0.41 ± 0.18	-8.56 ± 0.26 (M=1518) 0.52 ± 0.22
Divergent Blocks (no QED, 15k Budget) (N=4)	2866 ± 523	839 ± 1972	-10.57 ± 0.42 (M=1861) 0.32 ± 0.13	-9.48 ± 0.26 (M=3058) 0.40 ± 0.18	-8.55 ± 0.22 (M=2154) 0.48 ± 0.21

Configuration	**# Reaction Steps**	**# Unique Enforced Blocks**	**Oracle Budget (Wall Time)**
Divergent Blocks	3.68 ± 1.08 (M=6512)	1.75 ± 0.83	10,000 (8h 52m ± 42m)
Divergent Blocks (15k Budget)	3.61 ± 1.11 (M=15190)	1.80 ± 0.17	15,000 (15h 54m ± 1h 30m)
Divergent Blocks (no QED)	4.14 ± 1.36 (M=5397)	1 ± 0	10,000 (7h 39m ± 23m)
Divergent Blocks (no QED, 15k Budget)	4.30 ± 1.36 (M=8393)	1.75 ± 0.83	15,000 (12h 41m ± 25m)

G Retrosynthesis Model: Synthesis Routes

In this section, we show examples of synthetic routes from the MEGAN (Sacha et al., 2021) retrosynthesis model with enforced building blocks. The synthesis graph images were taken as is from Syntheseus' (Maziarz et al., 2023) output. Each figure in this section is from an experiment with a different enforced block set (100, 100 with "lucky" blocks purged, 10, 5, and divergent). Moreover, all routes will be shown for molecules with docking score < -10.5 since these are the most optimal. In addition, for the *5 Enforced Blocks*, the routes shown were from the runs without QED. This is because these were the only seeds that were successful under the oracle budget. **The enforced block is boxed.** We also try to show some diversity in the route lengths to highlight that path length was not explicitly optimized for. QED implicitly encourages shorter paths due to constraining the molecular weight, but even so, longer synthetic routes can still be observed (for example in Fig. 10). Future work could also reward shorter paths.



```
{'analysis_time': 0.0,
'first_solution_time': 0.022187,
'num_calls_rxn_model': 0,
'num_calls_value_function': 0,
'reaction_number_estimate': 0.0,
'retro_star_min_cost': 0.9017617490819452,
'retro_star_mol_cost': inf,
'retro_star_reaction_number': 0.9017617490819452,
'retro_star_value': 0.9017617490819452}
```

```
Reaction
{'analysis_time': 0.012164,
'first_solution_time': 0.022187,
'num_calls_rxn_model': 1,
'num_calls_value_function': 0,
'retro_star_min_cost': 0.9017617490819452,
'retro_star_reaction_number': 0.9017617490819452,
'retro_star_rxn_cost': 0.5772181333340674,
'retro_star_value': 0.9017617490819452}
```

```
{'analysis_time': 0.012164,
'first_solution_time': 0.022187,
'num_calls_rxn_model': 1,
'num_calls_value_function': 0,
'reaction_number_estimate': 0.0,
'retro_star_min_cost': 0.3245436157478779,
'retro_star_mol_cost': inf,
'retro_star_reaction_number': 0.3245436157478779,
'retro_star_value': 0.9017617490819452}
```

```
{'analysis_time': 0.012164,
'first_solution_time': 0.012164,
'num_calls_rxn_model': 1,
'num_calls_value_function': 0,
'retro_star_min_cost': 0.0,
'retro_star_mol_cost': 0.0,
'retro_star_reaction_number': 0.0,
'retro_star_value': 0.9017617490819452}
```

```
Reaction
{'analysis_time': 0.022187,
'first_solution_time': 0.022187,
'num_calls_rxn_model': 2,
'num_calls_value_function': 1,
'retro_star_min_cost': 0.3245436157478779,
'retro_star_reaction_number': 0.3245436157478779,
'retro_star_rxn_cost': 0.3245436157478779,
'retro_star_value': 0.9017617490819452}
```

```
{'analysis_time': 0.022187,
'first_solution_time': 0.022187,
'num_calls_rxn_model': 2,
'num_calls_value_function': 1,
'retro_star_min_cost': 0.0,
'retro_star_mol_cost': 0.0,
'retro_star_reaction_number': 0.0,
'retro_star_value': 0.9017617490819452}
```

```
{'analysis_time': 0.022187,
'first_solution_time': 0.022187,
'num_calls_rxn_model': 2,
'num_calls_value_function': 1,
'retro_star_min_cost': 0.0,
'retro_star_mol_cost': 0.0,
'retro_star_reaction_number': 0.0,
'retro_star_value': 0.9017617490819452}
```

```
{'analysis_time': 0.0,
'first_solution_time': 0.021941,
'num_calls_rxn_model': 0,
'num_calls_value_function': 0,
'reaction_number_estimate': 0.0,
'retro_star_min_cost': 0.5887689474620623,
'retro_star_mol_cost': inf,
'retro_star_reaction_number': 0.5887689474620623,
'retro_star_value': 0.5887689474620623}
```

```
Reaction
{'analysis_time': 0.01192,
'first_solution_time': 0.021941,
'num_calls_rxn_model': 1,
'num_calls_value_function': 0,
'retro_star_min_cost': 0.5887689474620623,
'retro_star_reaction_number': 0.5887689474620623,
'retro_star_rxn_cost': 0.4482367230842973,
'retro_star_value': 0.5887689474620623}
```

```
{'analysis_time': 0.01192,
'first_solution_time': 0.01192,
'num_calls_rxn_model': 1,
'num_calls_value_function': 0,
'retro_star_min_cost': 0.0,
'retro_star_mol_cost': 0.0,
'retro_star_reaction_number': 0.0,
'retro_star_value': 0.5887689474620623}
```

```
{'analysis_time': 0.01192,
'first_solution_time': 0.021941,
'num_calls_rxn_model': 1,
'num_calls_value_function': 0,
'reaction_number_estimate': 0.0,
'retro_star_min_cost': 0.14053222437776508,
'retro_star_mol_cost': inf,
'retro_star_reaction_number': 0.14053222437776508,
'retro_star_value': 0.5887689474620623}
```

```
Reaction
{'analysis_time': 0.021941,
'first_solution_time': 0.021941,
'num_calls_rxn_model': 2,
'num_calls_value_function': 1,
'retro_star_min_cost': 0.14053222437776508,
'retro_star_reaction_number': 0.14053222437776508,
'retro_star_rxn_cost': 0.14053222437776508,
'retro_star_value': 0.5887689474620623}
```

```
{'analysis_time': 0.021941,
'first_solution_time': 0.021941,
'num_calls_rxn_model': 2,
'num_calls_value_function': 1,
'retro_star_min_cost': 0.0,
'retro_star_mol_cost': 0.0,
'retro_star_reaction_number': 0.0,
'retro_star_value': 0.5887689474620623}
```

```
{'analysis_time': 0.021941,
'first_solution_time': 0.021941,
'num_calls_rxn_model': 2,
'num_calls_value_function': 1,
'retro_star_min_cost': 0.0,
'retro_star_mol_cost': 0.0,
'retro_star_reaction_number': 0.0,
'retro_star_value': 0.5887689474620623}
```

Figure 9: 100 Enforced Blocks example routes. The enforced block is boxed.

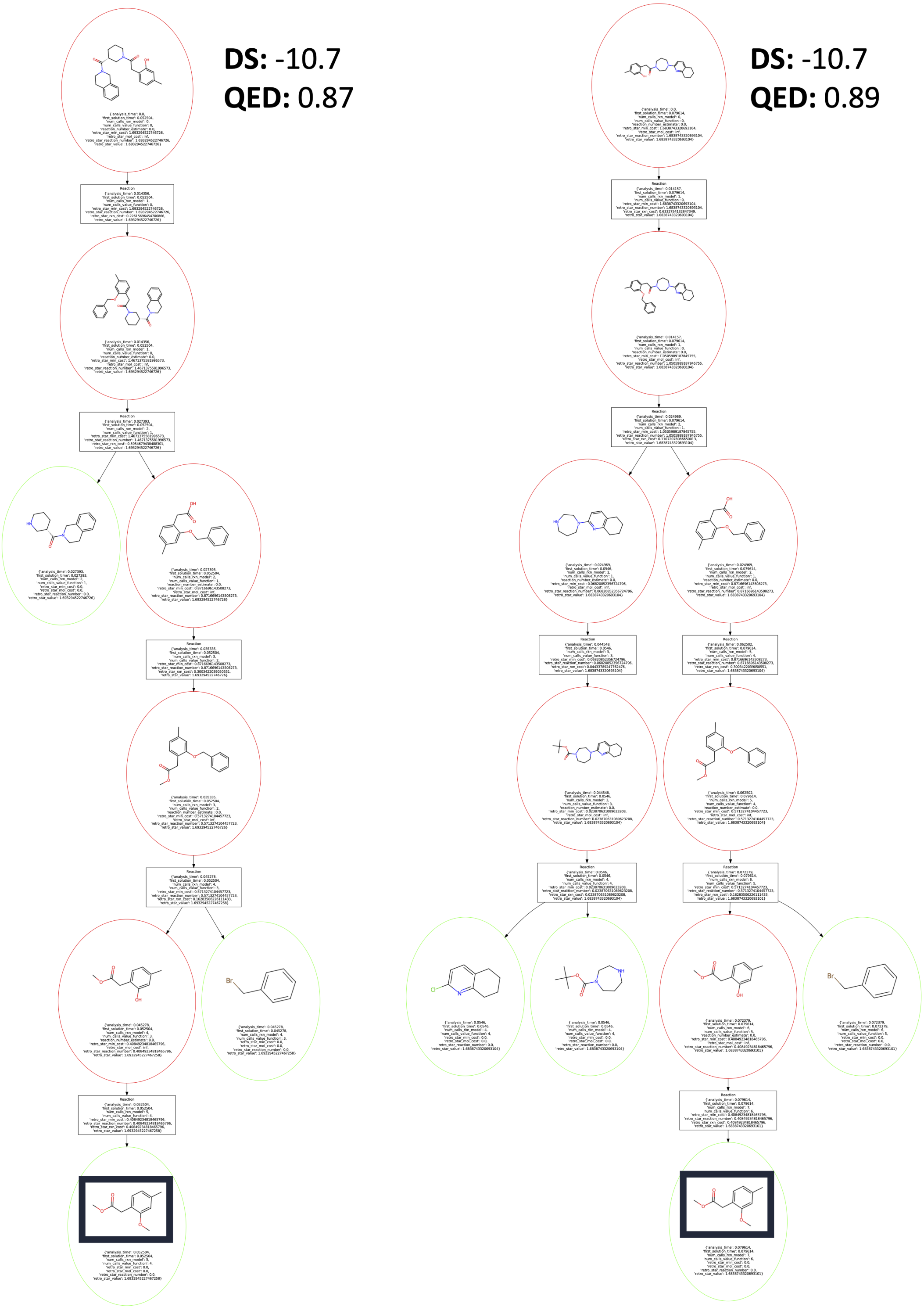


Figure 10: 100 Enforced Blocks Purged example routes. The enforced block is boxed.

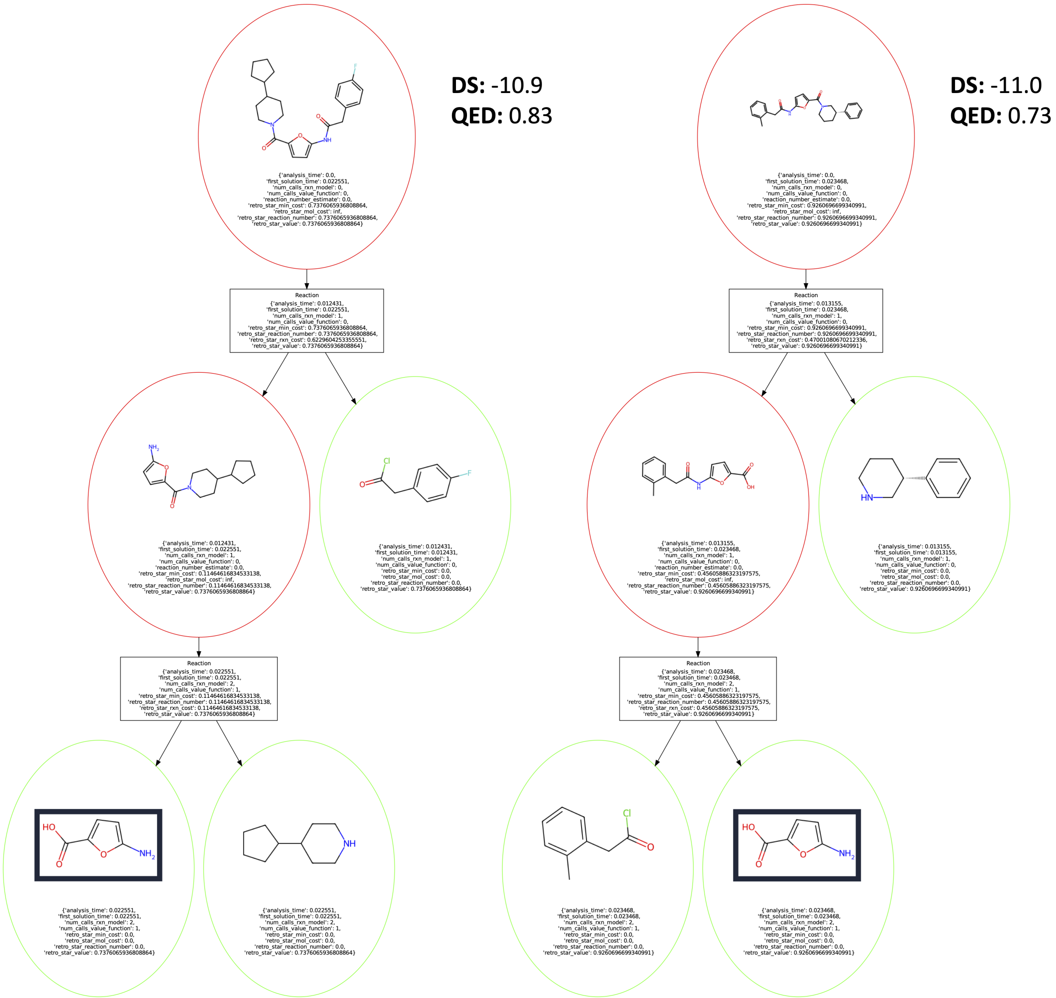

DS: -10.9
QED: 0.83
{'analysis_time': 0.0, 'first_solution_time': 0.022551, 'num_calls_rxn_model': 0, 'num_calls_value_function': 0, 'reaction_number_estimate': 0.0, 'retro_star_min_cost': 0.7376065936808864, 'retro_star_mol_cost': inf, 'retro_star_reaction_number': 0.7376065936808864, 'retro_star_value': 0.7376065936808864}
Reaction {'analysis_time': 0.012431, 'first_solution_time': 0.022551, 'num_calls_rxn_model': 1, 'num_calls_value_function': 0, 'retro_star_min_cost': 0.7376065936808864, 'retro_star_reaction_number': 0.7376065936808864, 'retro_star_rxn_cost': 0.6229604253355551, 'retro_star_value': 0.7376065936808864}
{'analysis_time': 0.012431, 'first_solution_time': 0.022551, 'num_calls_rxn_model': 1, 'num_calls_value_function': 0, 'reaction_number_estimate': 0.0, 'retro_star_min_cost': 0.11464616834533138, 'retro_star_mol_cost': inf, 'retro_star_reaction_number': 0.11464616834533138, 'retro_star_value': 0.7376065936808864}
{'analysis_time': 0.012431, 'first_solution_time': 0.012431, 'num_calls_rxn_model': 1, 'num_calls_value_function': 0, 'retro_star_min_cost': 0.0, 'retro_star_mol_cost': 0.0, 'retro_star_reaction_number': 0.0, 'retro_star_value': 0.7376065936808864}
Reaction {'analysis_time': 0.022551, 'first_solution_time': 0.022551, 'num_calls_rxn_model': 2, 'num_calls_value_function': 1, 'retro_star_min_cost': 0.11464616834533138, 'retro_star_reaction_number': 0.11464616834533138, 'retro_star_rxn_cost': 0.11464616834533138, 'retro_star_value': 0.7376065936808864}
{'analysis_time': 0.022551, 'first_solution_time': 0.022551, 'num_calls_rxn_model': 2, 'num_calls_value_function': 1, 'retro_star_min_cost': 0.0, 'retro_star_mol_cost': 0.0, 'retro_star_reaction_number': 0.0, 'retro_star_value': 0.7376065936808864}
{'analysis_time': 0.022551, 'first_solution_time': 0.022551, 'num_calls_rxn_model': 2, 'num_calls_value_function': 1, 'retro_star_min_cost': 0.0, 'retro_star_mol_cost': 0.0, 'retro_star_reaction_number': 0.0, 'retro_star_value': 0.7376065936808864}
DS: -11.0
QED: 0.73
{'analysis_time': 0.0, 'first_solution_time': 0.023468, 'num_calls_rxn_model': 0, 'num_calls_value_function': 0, 'reaction_number_estimate': 0.0, 'retro_star_min_cost': 0.9260696699340991, 'retro_star_mol_cost': inf, 'retro_star_reaction_number': 0.9260696699340991, 'retro_star_value': 0.9260696699340991}
Reaction {'analysis_time': 0.013155, 'first_solution_time': 0.023468, 'num_calls_rxn_model': 1, 'num_calls_value_function': 0, 'retro_star_min_cost': 0.9260696699340991, 'retro_star_reaction_number': 0.9260696699340991, 'retro_star_rxn_cost': 0.47001080670212336, 'retro_star_value': 0.9260696699340991}
{'analysis_time': 0.013155, 'first_solution_time': 0.023468, 'num_calls_rxn_model': 1, 'num_calls_value_function': 0, 'reaction_number_estimate': 0.0, 'retro_star_min_cost': 0.45605886323197575, 'retro_star_mol_cost': inf, 'retro_star_reaction_number': 0.45605886323197575, 'retro_star_value': 0.9260696699340991}
{'analysis_time': 0.013155, 'first_solution_time': 0.013155, 'num_calls_rxn_model': 1, 'num_calls_value_function': 0, 'retro_star_min_cost': 0.0, 'retro_star_mol_cost': 0.0, 'retro_star_reaction_number': 0.0, 'retro_star_value': 0.9260696699340991}
Reaction {'analysis_time': 0.023468, 'first_solution_time': 0.023468, 'num_calls_rxn_model': 2, 'num_calls_value_function': 1, 'retro_star_min_cost': 0.45605886323197575, 'retro_star_reaction_number': 0.45605886323197575, 'retro_star_rxn_cost': 0.45605886323197575, 'retro_star_value': 0.9260696699340991}
{'analysis_time': 0.023468, 'first_solution_time': 0.023468, 'num_calls_rxn_model': 2, 'num_calls_value_function': 1, 'retro_star_min_cost': 0.0, 'retro_star_mol_cost': 0.0, 'retro_star_reaction_number': 0.0, 'retro_star_value': 0.9260696699340991}
{'analysis_time': 0.023468, 'first_solution_time': 0.023468, 'num_calls_rxn_model': 2, 'num_calls_value_function': 1, 'retro_star_min_cost': 0.0, 'retro_star_mol_cost': 0.0, 'retro_star_reaction_number': 0.0, 'retro_star_value': 0.9260696699340991}

Figure 11: 10 Enforced Blocks example routes. The enforced block is boxed.

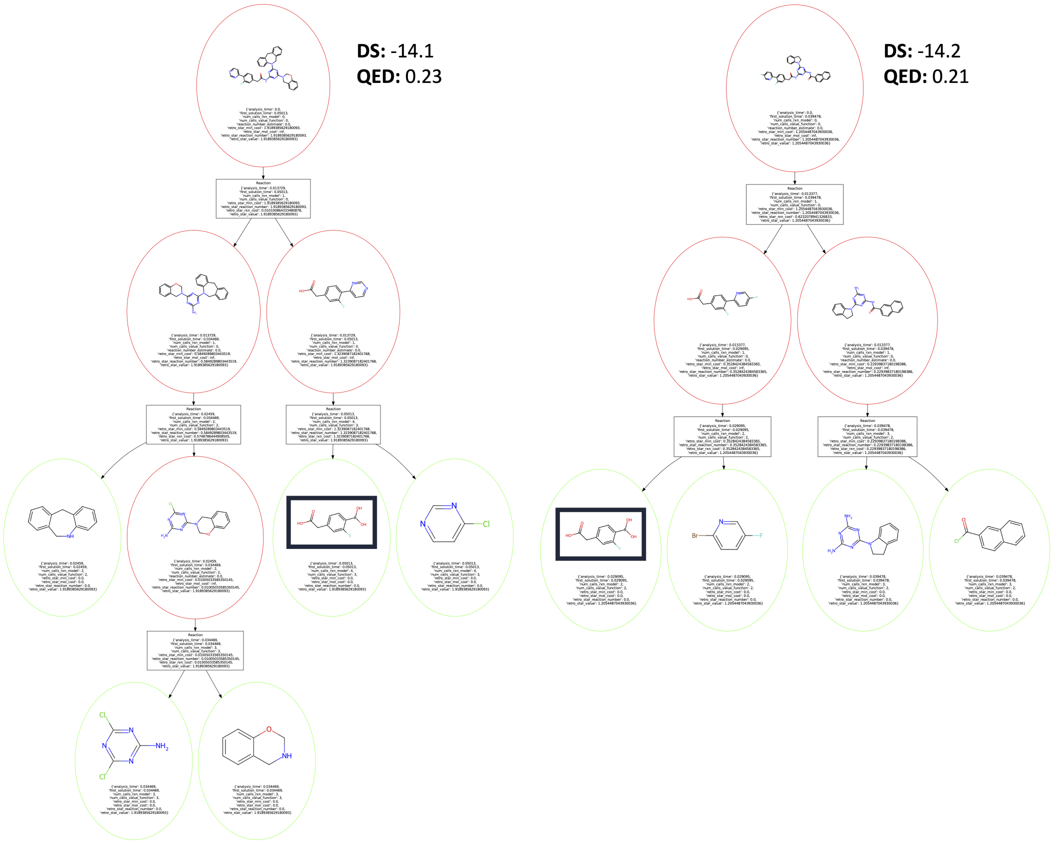


Figure 12: 5 Enforced Blocks example routes. Note that QED was not enforced here as the QED experiments did not successful generate any enforced blocks under the oracle budget. The enforced block is boxed.

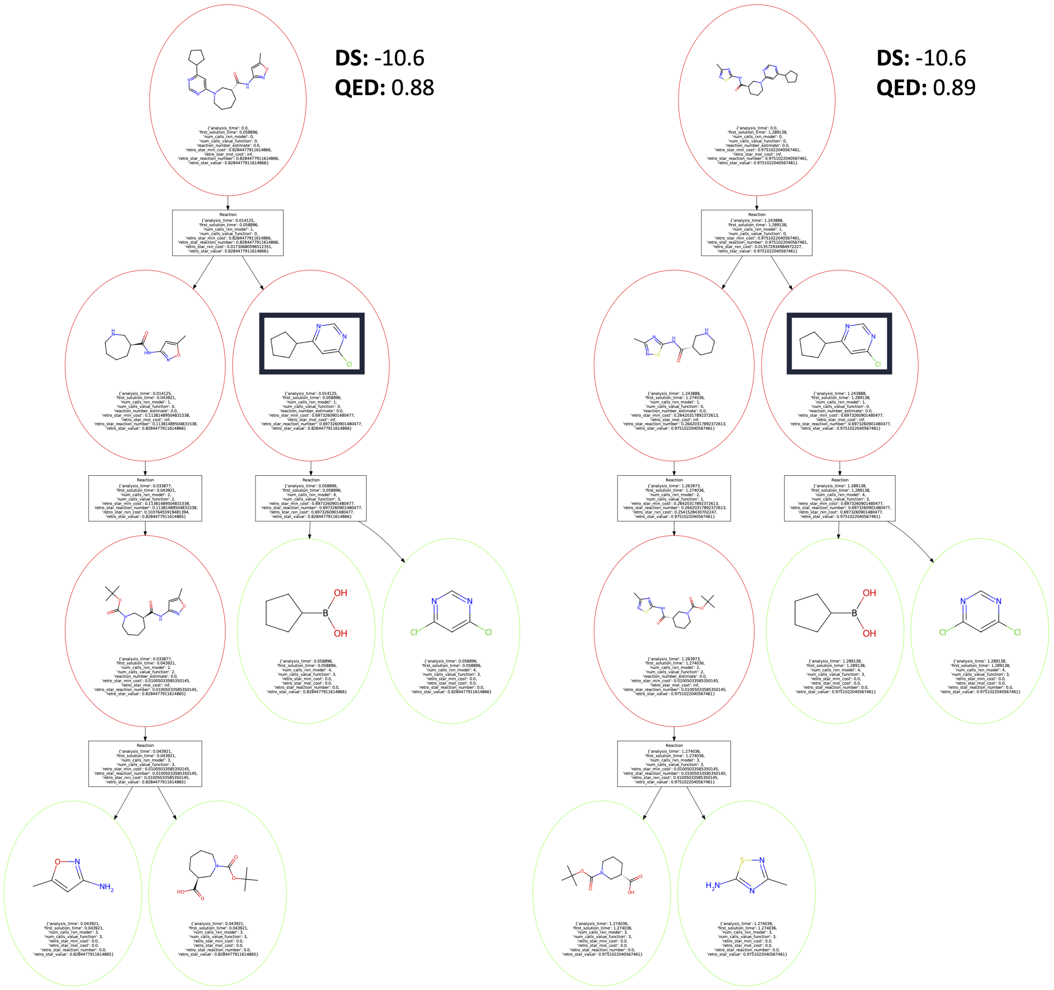


Figure 13: Divergent Enforced Blocks example routes. The enforced block is boxed.