SYNCOGEN: Synthesizable 3D Molecule Generation via Joint Reaction and Coordinate Modeling

Andrei Rekesh[1 2] Miruna Cretu[3] Dmytro Shevchuk[1] Vignesh Ram Somnath[4] Pietro Lio[3] Robert A. Batey[1] Mike Tyers[1 2] Michał Koziarski[1 2 5] Cheng-Hao Liu[6 7 8]

Abstract

Ensuring synthesizability remains a major challenge in generative small molecule design. While recent developments in synthesizable molecule generation have demonstrated promising results, they are largely confined to 2D molecular graph space, limiting their ability to perform geometry-based conditional generation. In this work, we present SYNCOGEN, a single framework with simultaneous masked graph diffusion and flow matching for synthesizable molecule generation. SYNCOGEN samples from the joint distribution of molecular building blocks, chemical reactions, and atomic coordinates. To train the model, we curated SYNSPACE, a dataset containing over 600K synthesis-aware building block graphs and 3.3M conformers. We show that SYNCOGEN achieves state-of-the-art performance on unconditional small molecule graph and conformer generation, and the model delivers competitive performance in zero-shot in linker design. Overall, our multimodal formulation represents a foundation for future applications enabled by non-autoregressive molecular generation, including analogue expansion, lead optimization, and direct structure conditioning.

1. Introduction

Generative models significantly enhance the efficiency of chemical space exploration in drug discovery by directly sampling molecules with desired properties. However, a key bottleneck in their practical deployment is low synthesizability—generated molecules are often difficult or impossible to produce in the laboratory (Gao & Coley, 2020). Addressing this limitation has recently become an active area of research. In particular, several template-based methods (Koziarski et al., 2024; Cretu et al., 2024; Seo et al., 2024; Gaiński et al., 2025; Gao et al., 2024; Jocys et al., 2024; Swanson et al., 2024) have been proposed to better reflect the chemical synthesis process. These approaches typically operate on representations akin to synthesis graphs, which abstract away the underlying 3D molecular structure. As a result, they are unable to access geometric information contained in molecular conformations when guiding generation toward molecules that exhibit desirable properties.

A promising alternative for molecular design involves spatial modeling at the atomic level. Inspired by advances in protein structure prediction (Yang et al., 2025; Campbell et al., 2024; Wang et al., 2025) and the development of generative frameworks such as diffusion and flow matching, recent work has focused on directly sampling 3D atomic coordinates of small molecules (Hassan et al., 2024; Jing et al., 2023; Fan et al., 2024). These methods learn to generate spatially meaningful, property-aligned conformations along with molecular graphs. The ability to model atomic structure directly increases the expressivity of these approaches, enabling applications such as pocket-conditioned generation (Lee & Cho, 2024), scaffold hopping (Torge et al., 2023; Yoo et al., 2024), molecular optimization (Morehead & Cheng, 2024), and analog discovery (Sun et al., 2025). However, integrating synthesizability constraints into these models remains a major challenge, and most existing 3D generative approaches do not address this issue.

This work aims to bridge the gap between 3D molecular generation and synthetic feasibility by introducing SYNCOGEN (Synthesizable Co-Generation), a generative model capable of directly sampling highly synthesizable molecules in 3D coordinate space (Figure 1). Our main contributions are as follows:

- **Generative Framework:** We propose a novel generative framework that combines masked graph diffusion with flow matching to jointly sample from the distribu-

[1]University of Toronto [2]The Hospital for Sick Children [3]University of Cambridge [4]ETH Zürich [5]Vector Institute [6]Mila - Quebec AI Institute [7]McGill University [8]Caltech. Correspondence to: Andrei Rekesh <a.rekesh@mail.utoronto.ca>, Cheng-Hao Liu <chenghao.liu@mail.mcgill.ca>.

Proceedings of the Workshop on Generative AI for Biology at the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

tion over building block-level reaction graphs and 3D atomic coordinates, unifying structure- and synthesis-aware modeling.

- **Molecular Dataset:** We curate and release a new dataset SynSpace comprising 622,766 synthesizable molecules represented as building block-level reaction graphs, along with 3,360,908 associated low-energy conformations. Compared to existing synthon-based datasets, ours enables the training of models that generate more readily synthesizable molecules and directly suggest streamlined synthetic routes.

- **Empirical Validation:** We demonstrate that SynCoGen outperforms existing methods in 3D molecule generation, all while modelling the reaction steps. As a multimodal model, SynCoGen can further predict conformations or generate discrete graphs. Lastly, SynCoGen performs zero-shot conditional molecular generation tasks such as linker design, highlighting its applicability for drug-discovery.

2. Background and Related Work

Flow Matching. Given two distributions ρ_0 and ρ_1, and an interpolating probability path ρ_t such that $\rho_{t=0} = \rho_0$ and $\rho_{t=1} = \rho_1$, flow matching (Lipman et al., 2023) aims to learn the underlying vector field u_t that generates ρ_t. Directly regressing u_t with a parametric version v_θ is unfortunately not possible as u_t is typically not known in closed form. Instead, flow matching defines a conditional probability path $\rho_{t|1}$ and its corresponding vector field $u_{t|1}$. The marginal vector field u_t can then be learnt with v_θ by regressing against $u_{t|1}$ with the CFM objective:

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t,\mathbf{x_1}\sim\rho_1,\mathbf{x}\sim\rho_{t|1}(\cdot|x_1)}||v_t(\mathbf{x};\theta) - u_{t|1}(\mathbf{x}|\mathbf{x_1})||^2 \tag{1}$$

Masked Discrete Diffusion Models. Consider a clean data-point $\mathbf{x} \sim \rho_{\text{data}}$, where $\mathbf{x}$ is represented as a one-hot encoding over K categories. Discrete diffusion models (Austin et al., 2021; Sahoo et al., 2024) map the complex data distribution ρ_{data} to a simpler distribution via a Markov process, with absorbing (or masked) diffusion being the most common. In the masked diffusion framework, the forward interpolation process $(\rho_t)_{t\in[0,1]}$ with the associated noise schedule $(\alpha_t)_{t\in[0,1]}$ results in marginals $q(\mathbf{z}_t|\mathbf{x}) = \text{Cat}(\mathbf{z}_t; \alpha_t\mathbf{x} + (1-\alpha_t)\mathbf{m})$, where $\mathbf{z}_t$ and $\mathbf{m}$ denote intermediate latent variables and the one-hot encoding for the special [MASK] token, respectively. The corresponding posterior can be derived as:

$$q(\mathbf{z}_s|\mathbf{z}_t, \mathbf{x}) = \begin{cases} \text{Cat}(\mathbf{z}_s; \mathbf{z}_t), & \mathbf{z}_t \neq \mathbf{m} \\ \text{Cat}(\mathbf{z}_s; \frac{(1-\alpha_t)\mathbf{m}+(\alpha_s-\alpha_t)\mathbf{x}}{1-\alpha_t}), & \mathbf{z}_t = \mathbf{m} \end{cases} \tag{2}$$

In the reverse process, one typically estimates $p_\theta(\mathbf{z}_s|\mathbf{z}_t)$, whose optimal form matches Equation (2):

$$p_\theta(\mathbf{z}_s|\mathbf{z}_t) = \begin{cases} \text{Cat}(\mathbf{z}_s; \mathbf{z}_t), & \mathbf{z}_t \neq \mathbf{m} \\ \text{Cat}(\mathbf{z}_s; \frac{(1-\alpha_t)\mathbf{m}+(\alpha_s-\alpha_t)\mathbf{x}_\theta(\mathbf{z_t},\mathbf{t})}{1-\alpha_t}), & \mathbf{z}_t = \mathbf{m} \end{cases} \tag{3}$$

Sahoo et al. (2024) include two modifications to $\mathbf{x}_\theta$, namely **zero-masking probabilities** and **carry-over unmasking**, leading to improved likelihoods. We also use this in this work.

3D Molecular Generation. Several recent works (Irwin et al., 2025; Le et al., 2023; Vignac et al., 2023; Huang et al., 2023; Dunn & Koes, 2024) tackle the problem of unconditional molecular structure and atomic co-generation by sampling from the joint distribution over atom types and coordinates. However, like all models that generate atomic coordinates directly, they lack the ability to constrain the design space to molecules accessible via synthetic chemistry. A concurrent work (Shen et al., 2025) recently explored the use of generated 3D structures to guide GFlowNet policies in designing the graph of synthon-based linear molecules, although it does not focus on the quality of the structures.

Synthesizable Molecule Generation. Beyond directly optimizing synthesizability scores (Liu et al., 2022; Guo & Schwaller, 2025) – which are often unreliable – the predominant approach to ensuring synthetic accessibility involves modifying generative models to incorporate reaction templates. Early methods explored autoencoders (Bradshaw et al., 2019; 2020), genetic algorithms (Gao et al., 2021), and reinforcement learning (Gottipati et al., 2020; Horwood & Noutahi, 2020). More recently, GFlowNet-based (Koziarski et al., 2024; Cretu et al., 2024; Seo et al., 2024; Gaiński et al., 2025) and transformer-based (Gao et al., 2024; Jocys et al., 2024) methods have gained prominence. Synthesizability-aware generative models have already shown practical utility in biological discovery tasks (Swanson et al., 2024). However, most existing methods only generate molecular graphs and do not produce 3D structures.

3. Dataset

Training a synthesizability-aware co-generation model requires a dataset of easily synthesizable molecules in an appropriate format. In addition to atomic coordinates, this includes a graph-based representation from which plausible synthetic pathways can be inferred. A common approach is to use synthons—theoretical structural units that can be combined to form complete molecules. While widely adopted (Baker et al., 2024; Grigg et al., 2025; Medel-Lacruz et al., 2025), synthon-based representations do not guarantee the existence of a valid synthesis route, and do not directly provide one even if it exists. Moreover, they lack the flexibility

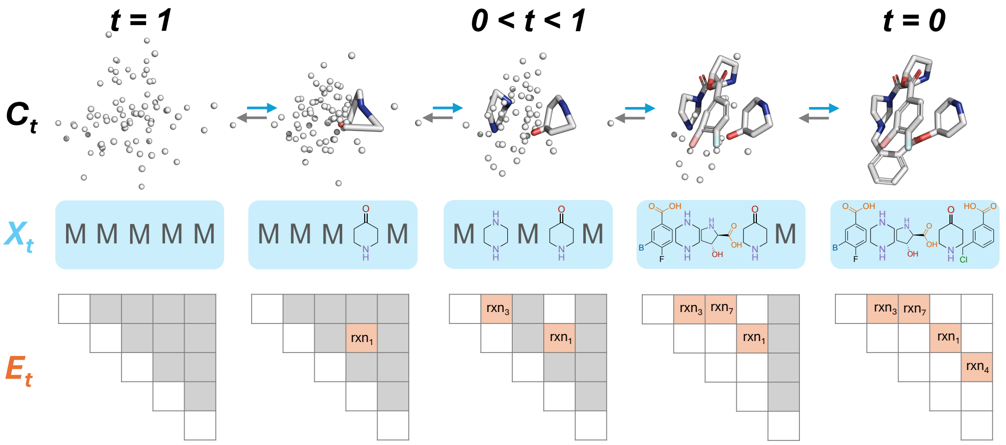

Figure 1. SYNCOGEN is a simultaneous masked graph diffusion and flow matching model that generates synthesizable molecules in 3D coordinate space. Each node corresponds to a building block and edges encode for chemical reactions. Note that nodes are not necessarily linear and that the leaving groups are not displayed.

to constrain the reaction space, which is often critical when prioritizing high-yield, high-reliability reactions or operating within the limits of automated synthesis platforms such as self-driving labs (Abolhasani & Kumacheva, 2023).

Alternatively, many synthesis-aware generators employ external reaction simulators, such as RDKit, to glue building blocks iteratively. While convenient, such black-box steps offer no fine-grained control when a reagent has multiple *reaction centers*, distinct atoms or atom sets that can each serve as the specific site of bond formation or cleavage in a coupling reaction. They also do not define atom–atom mappings between reactants and products, making it impossible to trace product atoms back to their parent building blocks, which in turn complicates edge assignment in building block graph generation.

To overcome these limitations, we curate a new dataset SYNSPACE comprising pairs of building block-level reaction graphs by constructing atom-level and building block-level graphs directly. We then calculate corresponding 3D coordinate tensors for each graph using semi-empirical methods (Bannwarth et al., 2019). See Figure 2 for an overview of the data creation process.

3.1. SYNSPACE: Graph Generation

We begin by constructing a vocabulary of 93 commercially available, low-cost building blocks and 19 high-yield reaction templates. This vocabulary is adapted from the collection proposed by Koziarski et al. (2024), retaining reactions that (1) ensure all product atoms originate from the two input reagents, and (2) involve at most one leaving group per reagent. We emphasize these are not only feasible chemistries, but rather simple and efficient reactions with readily-available building blocks that can enable rapid multi-synthesis.

We procedurally generate the SYNSPACE from this vocabulary by iteratively attaching building block graphs at their respective reaction centers with compatible chemical reaction templates, described in detail in Appendix A.2. Using this approach, we obtain 622,766 building block and reaction graphs, each constructed from 2 to 4 sequential reactions. For each resulting molecule, we generate multiple low-energy conformations and retain their atomic coordinates, with a total of 3,360,908 conformations. We will make the SYNSPACE publicly available.

Note: Injectivity. Many commercially available building blocks contain multiple reaction centers, each compatible with a different set of corresponding reaction centers on other building blocks. In this way, a building block-level reaction graph $G_b = (X, E)$ is not fully specified when edges are parametrized by the reaction alone. To achieve an injective correspondence, we therefore label edges from node i to $j > i$ by the triple

$$e_{ij} = (r, v_i, v_j), \tag{4}$$

where r is the coupling reaction and (v_i, v_j) are the participating reaction centers on the source and destination blocks, respectively. Strictly speaking, distinct stereoisomers that differ only in post-coupling chirality collapse to the same (X, E) representation, but this granularity suffices for the scope of the current work.

3.2. SYNSPACE: Conformation Generation

For each molecular graph, 50 initial conformers were generated with the ETKDG (Riniker & Landrum, 2015) algorithm (RDKit implementation). These structures were

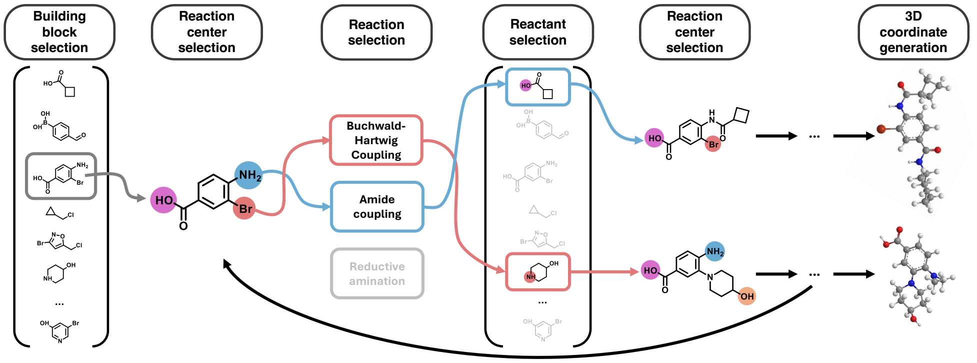

Figure 2. **Graphical overview of SynSpacecreation process.** Highly synthesizable molecules are procedurally constructed by sampling synthesis pathways from a predefined set of building blocks and reactions. Starting from an initial building block, a reaction center, a compatible reaction, and a suitable reactant are selected. This process is iteratively repeated for a fixed number of reaction steps. After the final structure is assembled, multiple low-energy 3D conformations are generated.

energy-minimised using the MMFF94 force field, and all conformers within 10 kcal/mol of the global minimum were retained. The resulting geometries were then re-optimised with the semi-empirical GFN2-xTB (Bannwarth et al., 2019) method, after which the same 10 kcal/mol energy threshold was applied. At every stage, redundant structures were removed by geometry-based clustering (RMSD < 1.5). This workflow yields, on average, 5.4 distinct conformers per graph. Relative to exhaustive approaches such as CREST (Pracht et al., 2024), the workflow is several orders of magnitude faster; despite occasionally omitting some conformations, the retained structures are diverse and reproduce the bond-length, bond-angle, and dihedral-angle distributions observed in CREST-derived datasets (see Section 5.1).

4. Methods

Notation. Let $\mathcal{B}$ be the building-block vocabulary and $\mathcal{R}$ the set of reaction templates, with cardinalities $B := |\mathcal{B}|$ and $R := |\mathcal{R}|$. We write N for the maximum number of building blocks that any molecule in the training set can contain, and M for the maximum number of atoms in a single building block. For each block $b \in \mathcal{B}$ we denote its set of reaction-center atoms by $\mathcal{V}(b)$; the global maximum of these counts is $V_{\max} := \max_{b\in\mathcal{B}} |\mathcal{V}(b)|$.

Hence tensor shapes contain factors such as $B + 1$ (to accommodate the masked token π_X in X), $RV_{\max}^2 + 2$ (to accommodate the no-edge and masked tokens λ_E and π_E), together with the bounds N and M introduced above. For any coordinate tensor C and binary mask S we define the mask-weighted centroid and its centered version by $\bar{C}_S := \frac{\sum S\odot C}{\sum S}$, $\quad \tilde{C} := C - \bar{C}_S$.

SynCoGen. SynCoGen is a masked-diffusion generator for building block-level reaction graphs and coordinates. Each molecule is represented by a triple (X, E, C) where $X \in \{0, 1\}^{N\times|\mathcal{B}|+1}$ encodes the sequence of building-block identities, $E \in \{0, 1\}^{N\times N\times|\mathcal{R}|V_{\max}^2+2}$ labels the coupling reaction (and centers) between every building block pair, and $C \in \mathbb{R}^{N\times M\times 3}$ stores all atomic coordinates. We describe the parameterization of graphs (X, E) in detail in Appendix B.1. Training combines two diffusion schemes: 1) a **discrete absorbing process** on (X, E) using the categorical forward kernel of Sahoo et al. (2024), and 2) a **continuous, visibility-aware process** on C whose endpoints are (i) a rototranslationally-aligned isotropic Gaussian and (ii) a recentered ground truth, considering all "visible" atoms in the prior.

In this section, we outline the most important components of the training and sampling procedures of SynCoGen. Specifically, this includes the model architecture (Section 4.1), noising schemes (Section 4.2), training-time constraints (Section 4.3), and sampling procedure (Section 4.4).

4.1. Model Architecture

At each timestep t, SynCoGen predicts building block logits L_t^X, L_t^E and a shifted coordinate estimate $\hat{\tilde{C}}_0^t$. The total loss is the weighted sum of the cross-entropy term $\mathcal{L}_{\text{graph}}$ on (X, E), the masked coordinate MSE term $\mathcal{L}_{\text{MSE}}$, and the short-range pairwise distance term $\mathcal{L}_{\text{pair}}$ (see Appendix B.3 for the training algorithm and Appendix B.11 for loss details). We implement a modified version of SemlaFlow (Irwin et al., 2025), a $SE(3)$ equivariant architecture originally designed for all-atom molecular design, as the principal backbone to generate both coordinates and graphs. As SemlaFlow handles atom and atom-level bond representations as both inputs and outputs, we define additional building-block-to-atom featurization protocols (Appendix B.2) and atom-to-building-block output layers (Appendix B.5).

4.2. Noising Schemes

Graph Noising. Following Sahoo et al. (2024), we corrupt true graphs (X_0, E_0) using the posterior described in Section 2. In practice, as all true edge matrices E_0 are symmetric, we symmetrize the sampled probabilities for the noising and denoising of E_t correspondingly (see Appendix B.6).

Coordinate Noising. For any time t where X_t contains a masked building block, we lack the information necessary to disregard any of its M possible atoms as padding. For this reason, we design a *visibility mask* S_t that considers all M atoms for each noised coordinate at time t as valid. We then center the prior by its visibility-masked centroid $\tilde{C}_1 = C_1 - \bar{C}_{1S_t}$. Here, all atoms $a \in \operatorname{supp}(S_t) \setminus \operatorname{supp}(S_0)$ are potentially valid at time t, but represent padding indices in the true molecule.

We thus must construct and interpolate a data-prior pair $(\tilde{C}_1, \tilde{C}_0)$ that contains a consistent number of valid atoms $|S_t|$ by which both $\tilde{C}_0$ and $\tilde{C}_1$ are centered. To handle atoms that do not appear in C_0, we record their points in $\tilde{C}_1$, re-center C_0 by the same visibility-masked centroid, then copy the atoms to their respective indices in C_0. This process is formalized in Section 4.2.

Here, $\mathcal{A}_i$ is the set of all atom indices a that constitute true atoms in X_0. Essentially, we task the model with rearranging the true atoms while disregarding padding by learning to fix padding atoms in place. Note that $S_t = S_0$ for all t where X_t contains no masked building blocks.

Algorithm 1 PAIRDATA(C_0, S_0, C_1, t, X_t)

Input: C_0 (clean coordinates), S_0 (atom mask), C_1 (prior sample), $t \in [0, 1]$, X_t (partially masked nodes)
Output: $\tilde{C}_0$ (re-centered ground truth), C_t (interpolated noisy coords)

```
$\mathcal{D}_t \leftarrow \{\, i \mid X_t[i] \neq \pi_X \,\}$    ▷ denoised blocks
$S_t[i, a] \leftarrow \mathbf{1}[i \notin \mathcal{D}_t \vee a \in \mathcal{A}_i]$    ▷ visibility
$\tilde{C}_1 \leftarrow C_1 - \bar{C}_{1S_t}$
$\tilde{C}_0 \leftarrow$ ZEROTENSOR()
for all $(i, a)$ do
    if $S_0[i, a] = 1$ then
        $\tilde{C}_0[i, a] \leftarrow C_0[i, a] - \bar{C}_{1S_t}$
    else if $S_t[i, a] = 1$ then    ▷ dummy atom
        $\tilde{C}_0[i, a] \leftarrow \tilde{C}_1[i, a]$
    end if
end for
$C_t \leftarrow (1 - t)\, \tilde{C}_0 + t\, \tilde{C}_1$
return $(\tilde{C}_0, C_t)$
```

Note: Non-Equivariance. This process results in data pairings where both C_0 and C_t are properly centered according to atoms that are possibly valid at time t. It is important to note that under this scheme, while the model is $SE(3)$-equivariant with respect to the system defined by the partial mask S_t, it is not equivariant with respect to the orientation of the molecule itself unless $\mathcal{D}_t^c = \varnothing$, as the presence and temporary validity of masked dummy atoms offsets the true atom centering and thus breaks both translational and rotational equivariance.

Flexible Atom Count. Most 3D *de novo* molecule generation methods requires the specification of the number of atoms. Because the prior of SYNCOGEN is over building blocks, we naturally handle flexible number of atoms during generation and model any excessive atoms as ghost atoms.

4.3. Training-time Constraints

SYNCOGEN inherits various training-time simplifications from MDLM (Sahoo et al., 2024), including zero masked logit probabilities and carry-over logit unmasking. In addition to these, we implement the following:

1. **No-Edge Diagonals.** We set the diagonals of all edge logit predictions L_θ^E to no-edge, as no building block has a coupling reaction-induced bond to itself.

2. **Edge Count Limit.** Let
$$k_t := \textstyle\sum_{1 \le i < j \le n} \mathbb{1}\big[E_t[i, j, \cdot] \notin \{\pi_E, \lambda_E\}\big], \quad (5)$$
be the number of unmasked true edges in the upper triangle of E_t. If $k_t = n - 1$, we have the correct number of edges for a molecule containing n building blocks and therefore set all remaining edge logits to λ_E.

3. **Compatibility Masking.** Assume that for some E_t an edge entry is already denoised, $E_t[i, j, \cdot] = (r, v_i, v_j)$, meaning that building block i reacts with building block j via reaction r and centers $v_i \in \mathcal{V}(X_i)$, $v_j \in \mathcal{V}(X_j)$. Define the sets of *center-matched reagents*
$$\begin{aligned} \mathcal{B}^A_{r,v} &:= \{\, b \in \mathcal{B} \mid (b, v) \text{ matches reagent A in } r \},\\ \mathcal{B}^B_{r,v} &:= \{\, b \in \mathcal{B} \mid (b, v) \text{ matches reagent B in } r \}. \end{aligned} \quad (6)$$
For every node slot i (resp. j) we construct a $|\mathcal{B}|$-dimensional binary mask
$$\begin{aligned} \mathcal{X}_{i,k} &= \mathbb{1}[b_k \in \mathcal{B}^A_{r,v_i}],\\ \mathcal{X}_{j,k} &= \mathbb{1}[b_k \in \mathcal{B}^B_{r,v_j}],\\ k &= 1, \dots, |\mathcal{B}|. \end{aligned} \quad (7)$$
so that the soft-max for $X_t[i, \cdot]$ (resp. $X_t[j, \cdot]$) is evaluated only over the 1-entries of $\mathcal{X}_i$ (resp. $\mathcal{X}_j$). Analogously, once a node identity $X_t[j] = b$ is denoised, incoming edge channels (i, j) with $j > i$ are masked to reactions $e = (r, v_i, v_j)$ such that $b \in \mathcal{B}^B_{r,v_i}$.

Put simply, we restrict logits to disallow loops, to impose the a limit on the number of edges, and to better ensure the selection of chemically compatible building blocks and reactions.

4.4. Sampling

Sampling begins by drawing a building block count $n \sim \mathrm{Cat}(\pi_{\text{frag}})$, setting the node and edge tensors to the masked tokens, $X_1[i,\cdot] = \pi_X,\ E_1[i,j,\cdot] = \pi_E$ for every $0 \le i,j < N$, and padding all $(i \ge n)$ rows/columns with the no–edge token λ_E. The initial coordinates are an isotropic Gaussian $C_1 \sim \mathcal{N}(0,I)^{N\times M\times 3}$. From this state the sampler walks backwards in diffusion time, and at each step it (i) recenters the current coordinates by the visibility mask S_t derived from X_t, (ii) generates node, edge and coordinate predictions with the trained model, (iii) applies the single-parent edge constraint (Section 4.4), (iv) draws the next discrete state, and (v) updates coordinates by an Euler step. After a deterministic pass at $t = 0$ we calculate $\big(\hat{X}_0, \hat{E}_0\big) = \arg\max_k L_\theta^E[\cdots, k]$ center the coordinates a final time, yielding the molecule $(\hat{X}_0, \hat{E}_0, \hat{C}_0)$. Complete line-by-line pseudocode is provided in Appendix B.4. Additionally, we find inference annealing (see Appendix D.1) to minorly improve performance at sampling time.

Constraint-Aware Edge Pruning. By construction, a molecule containing n connected building blocks contains exactly $n-1$ edges, and fragment $j > 0$ has a unique parent $i < j$. Consequently, let $E_\theta^t \in [0,1]^{n\times n\times|\mathcal{R}|V_{\max}^2}$ be the soft-max edge probabilities produced at step t. The routine below resolves the unique parent for every building block column $j > 0$ and returns a probability tensor $\tilde{E}_\theta^t$ with exactly one non–zero entry per column.

```
Algorithm 2 SampleEdges$\big(E_\theta^t, n\big)$
Input: edge probabilities $E_\theta^t$
Output: pruned probabilities $\tilde{E}_\theta^t$
$\tilde{E}_\theta^t \leftarrow \mathbf{0}$
for $j = 1$ to $n-1$ do
    $(i_j, e_j) \sim \mathrm{Cat}\big(\{E_\theta^t[i,j,e] \mid 0 \le i < j\}\big)$
    $\tilde{E}_\theta^t[i_j, j, e_j] \leftarrow 1$
end for
return $\tilde{E}_\theta^t$
```

$\tilde{E}_\theta^t$ is then symmetrized and fed to the discrete reverse sampler described in Appendix B.6.

5. Experiments

5.1. *De Novo* 3D Molecule Generation

We first study SynCoGen in unconditional molecule generation jointly with 3D coordinates and reaction graphs. We evaluate SynCoGen against several recently published all-atom generation frameworks which produce 3D coordinates, including SemlaFlow (Irwin et al., 2025), EQGAT-Diff (Le et al., 2023), MiDi (Vignac et al., 2023), JODO (Huang et al., 2023), and FlowMol (Dunn & Koes, 2024). To isolate modelling from data effects, we retrain SemlaFlow on SynSpacefor the same number of epochs as SynCoGen.

For each model, we sample 1000 molecules and compute stringent metrics capturing chemical soundness, synthetic accessibility, conformers quality, and distributional fidelity. Pertaining to the molecular graph, we report the RDKit sanitization validity (Valid.) and retrosynthetic solve rate (AiZynthFinder (Genheden et al., 2020) (AiZyn.) and Syntheseus (Maziarz et al., 2025) (Synth.)). For the generated conformers, we compute the median non-bonded interaction energies per atom via the forcefield method GFN-FF and the semiempirical quantum chemistry method GFN2-xTB (Bannwarth et al., 2019; Spicher & Grimme, 2020), as well as PoseBusters (Buttenschoen et al., 2024) validity rate (PB Vali.). We also evaluate the diversity (Div.) as average pairwise Tanimoto similarity, novelty (Nov.) as the percentages of candidates not appearing in the training set, and the Fréchet ChemNet Distance (Preuer et al., 2018) (FCD) on the distance between generated samples and the training distribution. Details on how the metrics are computed can be found in the Appendix.

The results are presented in Table 1. For chemical reasonableness, we see that SynCoGen generates almost entirely valid molecules. Our generation process mimics a multi-step reaction pathway, and as a result our molecules are significantly more synthesizable compared to baseline methods. Because AiZynthFinder and Syntheseus solve only 50–70 % of known drug-like molecules, our 50–72 % scores likely underestimate true synthesizability.

For structural reasonableness, the generated conformers reproduce the overall energy distributions and have very favorable non-covalent interaction energies as evaluated by relatively accurate computational chemistry methods, especially when compared to the baseline methods (Table 1 and Figure 3). This is also evident by the lack of structural changes upon further geometric relaxation (Figure 9). The low non-bonded energies indicate SynCoGen learns to sample many intramolecular interactions, also seen with our example samples (Figure 8). Quantitatively, 87% of these conformers pass PoseBusters pose plausibility checks. Furthermore, SynCoGen reproduce the delicate data distribution of bond lengths, angles, and dihedrals (Figures 3

Table 1. **Comparison of generative methods.** ↑/↓ indicate that higher/lower is better.

		Primary metrics					**Secondary metrics**			
Group	**Method**	Valid. ↑	AiZyn. ↑	Synth. ↑	GFN-FF ↓	xTB-2 ↓	PB Val. ↑	FCD ↓	Div. ↑	Nov. ↑
Rxns & coords	SynCoGen	**96.7**	**50**	**72**	**3.01**	**-0.91**	**87.2**	**2.91**	0.78	93.9
Atoms & coords	SemlaFlow	93.3	38	36	5.96	-0.72	**87.2**	7.21	0.85	99.6
	SemlaFlow w. SynSpace	72.0	27	48	3.27	-0.80	60.3	2.95	0.80	93.0
	EQGAT-diff	85.9	37	24	4.89	-0.73	78.9	6.75	**0.86**	99.5
	MiDi	74.4	33	31	4.90	-0.74	63.0	6.00	0.85	99.6
	JODO	91.1	38	31	4.72	-0.74	84.1	4.22	0.85	99.4
	FlowMol-CTMC	89.5	24	25	5.91	-0.68	69.3	13.0	**0.86**	**99.8**
	FlowMol-Gaussian	48.3	6	8	4.24	-0.71	30.7	21.0	**0.86**	99.7

Table 2. **Ablations studies.** We incrementally remove inference annealing, auxiliary losses, self-conditioning, scaled-noise, and constraints to see the performance difference. All results shown are at 50 epochs rather than 100 epochs in Table 1. See Sections 4.3 and 4.4, (Appendices B.3, B.9 and B.11).

Method	Valid. ↑	GFN-FF ↓
Base	93.5	4.871
- Inference annealing	93.5	4.933
- Auxiliary losses	85.3	5.194
- Self-conditioning	69.0	6.424
- Scaled noise	70.4	5.091
- Constraints	42.4	67.006

and 7). For example, SynCoGen generates fewer sp^2C-sp^2N bonds that are too short, captures peaky bond angle distributions (sp^3C-sp^3C-sp^3N), and replicate the broad dihedral angle distribution of the flexible sp^3C-sp^3C-sp^3C-sp^3C while still covering the rigid dihedral angles (e.g., sp^3C-sp^2C-sp^2C-sp^2C). Wasserstein-1 distances and JSD can be found in Appendix D.3, and additional plots on other bond lengths/angles/dihedrals can be found in Figure 7.

Training-, sampling-time and architectural ablations are presented in Table 2 and Appendix D.1. The largest performance enhancement originates froom constraints and self-conditioning with other training/sampling details also contributing. The performance gap between SynCoGen and SemlaFlow retrained on our dataset underscores that our training procedure — rather than the architecture or dataset — is the primary driver of performance. The multimodal can further perform other tasks; for example, given randomly assembled molecules from the reaction graph, SynCoGen can perform zero-shot conformer generation at a quality similar to ETKDG as implemented in RDKit (Table 4).

Finally, SynCoGen captures the training distribution as indicated by the low FCD, while generally producing novel molecules. The generated samples have slightly lower diversity as a trade-off of using a (limited) set of reaction building blocks. All generated samples are unique.

5.2. Molecular Inpainting for Fragment Linking

To demonstrate the practical usefulness of SynCoGen, we study fragment linking (Bancet et al., 2020) to design easily-synthesizable analogues of hard-to-make drugs. Fragment linking in drug design enables the construction of potent molecules by connecting smaller fragments that are known to bind distinct regions of a target site. We formulate fragment linking as a molecular inpainting task, where we fix the identity and coordinates of two fragments in a known ligand and sample its missing parts consistent with both geometry and reaction grammar.

As a case study, we pick several FDA-approved, hard-to-synthesize small-molecules with experimental crystal structures, each bound to a different target protein. These ligands contains a match for at least two of our building blocks. At sampling time, we condition on the substructure match by keeping fixed fragments denoised and replacing their coordinates with the corresponding linearly interpolated configuration at each time step t (see Appendix B.13 for details).

We evaluate our generated molecules using AutoDock Vina (Figure 4). SynCoGen consistently produces molecules with docking scores on par with or better than the native ligand while satisfying constraints on the presence of specific building blocks. Crucially, unlike existing approaches, it also guarantees a streamlined synthetic route by design (Table 6). Sample synthetic pathways are shown in Figure 10. The proof-of-concept inpainting setup is directly applicable for tasks such as scaffold hopping or the generation of synthesizable analogs.

Lastly, we emphasize using SynCoGen in this way does not require any retraining, and unlike previous methods (Schneuing et al., 2024; Igashov et al., 2024) the model links fragments using building blocks and reactions to ensure the synthesizability of the designs. We benchmarked SynCoGen against the state-of-the-art, purpose-built fragment-linking model DiffLinker (Igashov et al., 2024), and the

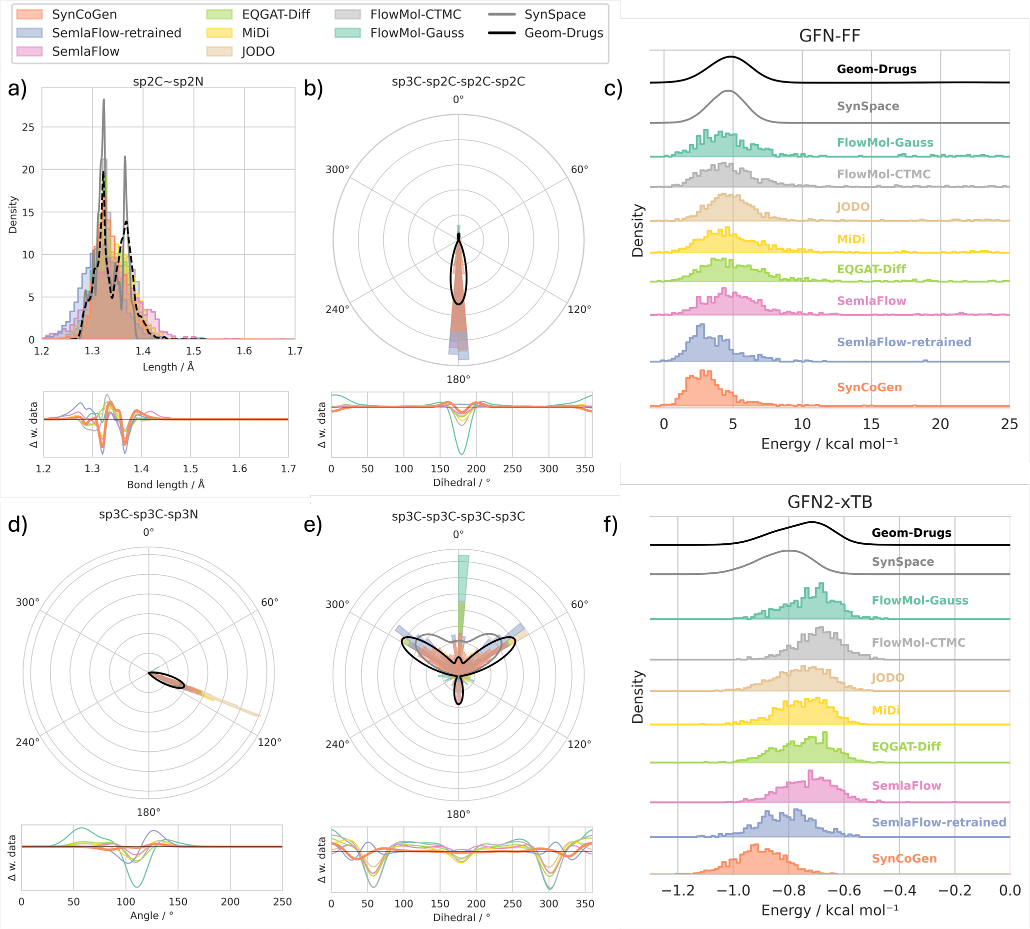

Figure 3. **Conformer energy and angle and bond length distribution comparisons.** a) Bond lengths, b) bond angles, c) and d) dihedral angles, e) average per-atom GFN2-xTB non-covalent interaction energies and f) average per-atom GFN-FF non-bonded interaction energies. Comparison to baseline generative models and the kernel density estimation of the training data are shown as differences or as solid lines.

retrosynthesis solve rate for DiffLinker is 0% where as it is 58-79% for SYNCOGEN (Table 6) .

6. Conclusion

In this work, we introduced SYNCOGEN, a generative model for synthesizable 3D molecular design. To support training, we curated a new dataset of highly synthesizable molecules paired with low-energy 3D conformations. SYNCOGEN achieves state-of-the-art performance on standard 3D generation benchmarks, while uniquely enabling direct reconstruction of synthesis pathways. We also showed its practical utility in molecular inpainting for easily-synthesizable analog design and scaffold hopping.

While SYNCOGEN marks a significant step toward synthesizability-aware 3D generation, it also opens several directions for future work. Currently, the model is trained unconditionally to produce valid conformations. Future extensions could incorporate property-conditioned generation—either through guided optimization or direct conditioning on binding pockets to design high-affinity ligands. One limitation of the current approach is the relatively constrained set of reactions and building blocks, which could be expanded to cover a broader chemical space. Nonetheless, curated reaction vocabularies may remain advantageous in contexts like automated or high-throughput synthesis.

References

Abolhasani, M. and Kumacheva, E. The rise of self-driving labs in chemical and materials sciences. *Nature Synthesis*, 2(6):483–492, 2023.

Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A. J., Bambrick, J., Bodenstein, S. W., Evans, D. A., Hung, C.-C., O'Neill, M., Reiman, D., Tunyasuvunakool, K., Wu,

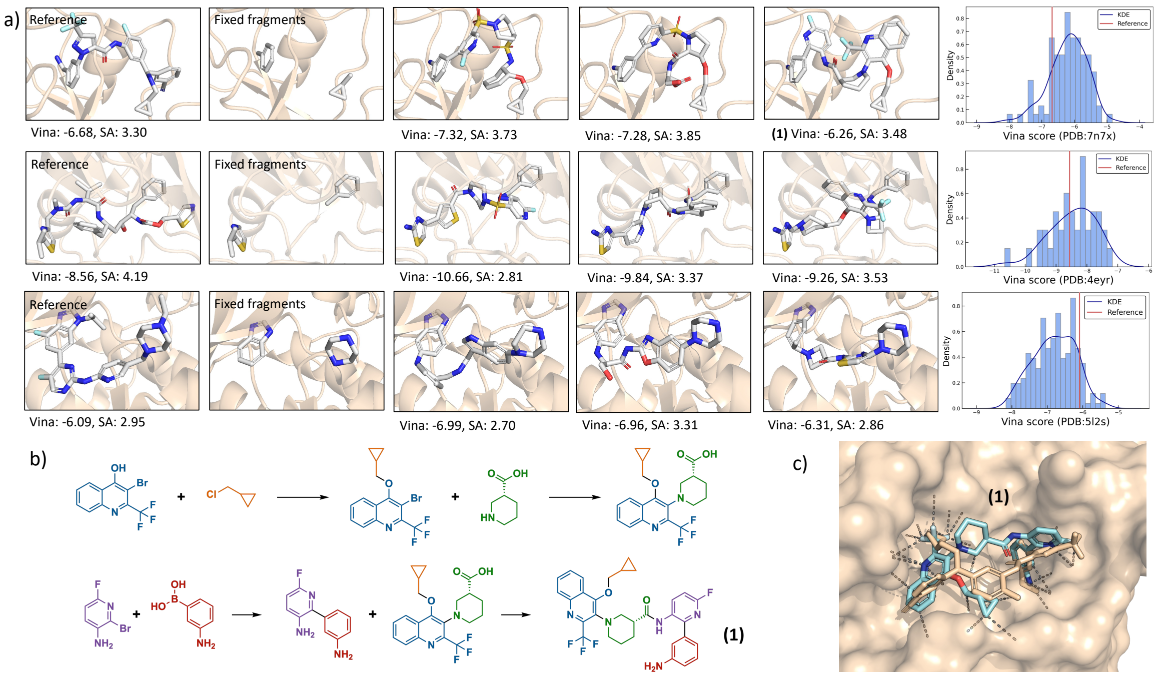

Figure 4. **Molecular inpainting.** a) We perform fragment linking starting from three experimentally identified ligands in the PDB. Starting fragments are substructure matches with building blocks used by our model. We show three examples of linkers generated by SYNCOGEN per structure and their computed Vina scores. b) Proposed synthesis pathway for molecule **(1)** sampled from our model and c) structure of **(1)** docked onto PDB 7n7x using AlphaFold3 compared against the PDB ligand.

Z., Žemgulytė, A., Arvaniti, E., Beattie, C., Bertolli, O., Bridgland, A., Cherepanov, A., Congreve, M., Cowen-Rivers, A. I., Cowie, A., Figurnov, M., Fuchs, F. B., Gladman, H., Jain, R., Khan, Y. A., Low, C. M. R., Perlin, K., Potapenko, A., Savy, P., Singh, S., Stecula, A., Thillaisundaram, A., Tong, C., Yakneen, S., Zhong, E. D., Zielinski, M., Žídek, A., Bapst, V., Kohli, P., Jaderberg, M., Hassabis, D., and Jumper, J. M. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, 630(8016):493–500, June 2024.

Austin, J., Johnson, D. D., Ho, J., Tarlow, D., and Van Den Berg, R. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.

Axelrod, S. and Gomez-Bombarelli, R. Geom, energy-annotated molecular conformations for property prediction and molecular generation. *Scientific Data*, 9(1):185, 2022.

Baker, F. N., Chen, Z., Adu-Ampratwum, D., and Ning, X. RLSynC: Offline–online reinforcement learning for synthon completion. *Journal of Chemical Information and Modeling*, 64(17):6723–6735, 2024.

Bancet, A., Raingeval, C., Lomberget, T., Le Borgne, M., Guichou, J.-F., and Krimm, I. Fragment linking strategies for structure-based drug design. *Journal of medicinal chemistry*, 63(20):11420–11435, 2020.

Bannwarth, C., Ehlert, S., and Grimme, S. GFN2-xTB—An accurate and broadly parametrized Self-Consistent Tight-Binding quantum chemical method with multipole electrostatics and Density-Dependent dispersion contributions. *J. Chem. Theory Comput.*, 15(3):1652–1671, March 2019.

Bose, A. J., Akhound-Sadegh, T., Huguet, G., Fatras, K., Rector-Brooks, J., Liu, C.-H., Nica, A. C., Korablyov, M., Bronstein, M., and Tong, A. Se(3)-stochastic flow matching for protein backbone generation, 2024. URL https://arxiv.org/abs/2310.02391.

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

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

Buttenschoen, M., Morris, G. M., and Deane, C. M. Posebusters: Ai-based docking methods fail to generate physically valid poses or generalise to novel sequences. *Chemical Science*, 15(9):3130–3139, 2024. ISSN 2041-6539. doi: 10.1039/d3sc04185a. URL http://dx.doi.org/10.1039/D3SC04185A.

Campbell, A., Yim, J., Barzilay, R., Rainforth, T., and Jaakkola, T. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design, 2024. URL https://arxiv.org/abs/2402.04997.

Cretu, M., Harris, C., Roy, J., Bengio, E., and Liò, P. SynFlowNet: Towards molecule design with guaranteed synthesis pathways. *arXiv preprint arXiv:2405.01155*, 2024.

Dunn, I. and Koes, D. R. Mixed continuous and categorical flow matching for 3d de novo molecule generation, 2024. URL https://arxiv.org/abs/2404.19739.

Fan, Z., Yang, Y., Xu, M., and Chen, H. EC-Conf: A ultra-fast diffusion model for molecular conformation generation with equivariant consistency. *Journal of Cheminformatics*, 16(1):107, September 2024.

Flamary, R., Courty, N., Gramfort, A., Alaya, M. Z., Boisbunon, A., Chambon, S., Chapel, L., Corenflos, A., Fatras, K., Fournier, N., Gautheron, L., Gayraud, N. T., Janati, H., Rakotomamonjy, A., Redko, I., Rolet, A., Schutz, A., Seguy, V., Sutherland, D. J., Tavenard, R., Tong, A., and Vayer, T. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021. URL http://jmlr.org/papers/v22/20-451.html.

Gaiński, P., Boussif, O., Shevchuk, D., Rekesh, A., Parviz, A., Tyers, M., Batey, R. A., and Koziarski, M. Scalable and cost-efficient de novo template-based molecular generation. In *ICLR 2025 Workshop on Generative and Experimental Perspectives for Biomolecular Design*, 2025.

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

Gao, W., Mercado, R., and Coley, C. W. Amortized tree generation for bottom-up synthesis planning and synthesizable molecular design. *arXiv preprint arXiv:2110.06389*, 2021.

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

Genheden, S., Thakkar, A., Chadimová, V., Reymond, J.-L., Engkvist, O., and Bjerrum, E. AiZynthFinder: a fast, robust and flexible open-source software for retrosynthetic planning. *Journal of Cheminformatics*, 12(1):70, November 2020.

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

Grigg, T. G., Burlage, M., Scott, O. B., Sydow, D., and Wilbraham, L. Active learning on synthons for molecular design. In *ICLR 2025 Workshop on Generative and Experimental Perspectives for Biomolecular Design*, 2025.

Guo, J. and Schwaller, P. Directly optimizing for synthesizability in generative molecular design using retrosynthesis models. *Chemical Science*, 16(16):6943–6956, 2025.

Hassan, M., Shenoy, N., Lee, J., Stark, H., Thaler, S., and Beaini, D. Et-flow: Equivariant flow-matching for molecular conformer generation. 2024. URL https://arxiv.org/abs/2410.22388.

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

Huang, H., Sun, L., Du, B., and Lv, W. Learning joint 2d & 3d diffusion models for complete molecule generation, 2023. URL https://arxiv.org/abs/2305.12347.

Igashov, I., Stärk, H., Vignac, C., Schneuing, A., Satorras, V. G., Frossard, P., Welling, M., Bronstein, M., and Correia, B. Equivariant 3d-conditional diffusion model for molecular linker design. *Nature Machine Intelligence*, 2024.

Irwin, R., Tibo, A., Janet, J. P., and Olsson, S. Semlaflow – efficient 3d molecular generation with latent attention and equivariant flow matching, 2025. URL https://arxiv.org/abs/2406.07266.

Jing, B., Corso, G., Chang, J., Barzilay, R., and Jaakkola, T. Torsional diffusion for molecular conformer generation. 2023. URL https://arxiv.org/abs/2206.01729.

Jocys, Z., Willems, H. M., and Farrahi, K. SynthFormer: Equivariant pharmacophore-based generation of molecules for ligand-based drug design. *arXiv preprint arXiv:2410.02718*, 2024.

Koziarski, M., Rekesh, A., Shevchuk, D., van der Sloot, A., Gaiński, P., Bengio, Y., Liu, C., Tyers, M., and Batey, R. RGFN: Synthesizable molecular generation using GFlowNets. *Advances in Neural Information Processing Systems*, 37:46908–46955, 2024.

Le, T., Cremer, J., Noé, F., Clevert, D.-A., and Schütt, K. Navigating the design space of equivariant diffusion-based generative models for de novo 3d molecule generation, 2023. URL https://arxiv.org/abs/2309.17296.

Lee, D. and Cho, Y. Fine-tuning pocket-conditioned 3D molecule generation via reinforcement learning. In *ICLR 2024 Workshop on Generative and Experimental Perspectives for Biomolecular Design*, 2024.

Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling, 2023. URL https://arxiv.org/abs/2210.02747.

Liu, C.-H., Korablyov, M., Jastrzebski, S., Włodarczyk-Pruszynski, P., Bengio, Y., and Segler, M. Retrognn: fast estimation of synthesizability for virtual screening and de novo design by learning from slow retrosynthesis software. *Journal of Chemical Information and Modeling*, 62(10):2293–2300, 2022.

Maziarz, K., Tripp, A., Liu, G., Stanley, M., Xie, S., Gaiński, P., Seidl, P., and Segler, M. H. S. Re-evaluating retrosynthesis algorithms with syntheseus. *Faraday Discussions*, 256:568–586, 2025. ISSN 1364-5498. doi: 10.1039/d4fd00093e. URL http://dx.doi.org/10.1039/D4FD00093E.

Medel-Lacruz, B., Herrero, A., Martín, F., Herrero, E., Luque, F. J., and Vázquez, J. Synthon-based strategies exploiting molecular similarity and protein–ligand interactions for efficient screening of ultra-large chemical libraries. *Journal of Chemical Information and Modeling*, 2025.

Morehead, A. and Cheng, J. Geometry-complete diffusion for 3D molecule generation and optimization. *Communications Chemistry*, 7(1):150, 2024.

Pracht, P., Grimme, S., Bannwarth, C., Bohle, F., Ehlert, S., Feldmann, G., Gorges, J., Müller, M., Neudecker, T., Plett, C., Spicher, S., Steinbach, P., Wesołowski, P. A., and Zeller, F. Crest—a program for the exploration of low-energy molecular chemical space. *The Journal of Chemical Physics*, 160(11):114110, 03 2024. ISSN 0021-9606. doi: 10.1063/5.0197592. URL https://doi.org/10.1063/5.0197592.

Preuer, K., Renz, P., Unterthiner, T., Hochreiter, S., and Klambauer, G. Fréchet ChemNet distance: A metric for generative models for molecules in drug discovery. *J. Chem. Inf. Model.*, 58(9):1736–1741, September 2018.

Riniker, S. and Landrum, G. A. Better informed distance geometry: Using what we know to improve conformation generation. *J. Chem. Inf. Model.*, 55(12):2562–2574, December 2015.

Sahoo, S. S., Arriola, M., Schiff, Y., Gokaslan, A., Marroquin, E., Chiu, J. T., Rush, A., and Kuleshov, V. Simple and effective masked diffusion language models, 2024. URL https://arxiv.org/abs/2406.07524.

Schneuing, A., Harris, C., Du, Y., Didi, K., Jamasb, A., Igashov, I., Du, W., Gomes, C., Blundell, T., Lio, P., Welling, M., Bronstein, M., and Correia, B. Structure-based drug design with equivariant diffusion models, 2024. URL https://arxiv.org/abs/2210.13695.

Seo, S., Kim, M., Shen, T., Ester, M., Park, J., Ahn, S., and Kim, W. Y. Generative flows on synthetic pathway for drug design. *arXiv preprint arXiv:2410.04542*, 2024.

Shen, T., Seo, S., Irwin, R., Didi, K., Olsson, S., Kim, W. Y., and Ester, M. Compositional flows for 3D molecule and synthesis pathway co-design. *arXiv preprint arXiv:2504.08051*, 2025.

Spicher, S. and Grimme, S. Robust atomistic modeling of materials, organometallic, and biochemical systems. *Angewandte Chemie International Edition*, 59(36):15665–15673, 2020.

Sun, M., Lo, A., Guo, M., Chen, J., Coley, C. W., and Matusik, W. Procedural synthesis of synthesizable molecules. In *The Thirteenth International Conference on Learning Representations*, 2025.

Swanson, K., Liu, G., Catacutan, D. B., Arnold, A., Zou, J., and Stokes, J. M. Generative AI for designing and validating easily synthesizable and structurally novel antibiotics. *Nature Machine Intelligence*, 6(3):338–353, 2024.

Torge, J., Harris, C., Mathis, S. V., and Lio, P. DiffHopp: A graph diffusion model for novel drug design via scaffold hopping. *arXiv preprint arXiv:2308.07416*, 2023.

Vignac, C., Osman, N., Toni, L., and Frossard, P. Midi: Mixed graph and 3d denoising diffusion for molecule generation, 2023. URL https://arxiv.org/abs/2302.09048.

Wang, C., Alamdari, S., Domingo-Enrich, C., Amini, A. P., and Yang, K. K. Toward deep learning sequence–structure co-generation for protein design. *Current Opinion in Structural Biology*, 91:103018, 2025. ISSN 0959-440X. doi: https://doi.org/10.1016/j.sbi.2025.103018. URL https://www.sciencedirect.com/science/article/pii/S0959440X25000363.

Yang, S., Ju, L., Peng, C., Zhou, J., Cai, Y., and Feng, D. Co-design protein sequence and structure in discrete space via generative flow. *Bioinformatics*, pp. btaf248, 04 2025. ISSN 1367-4811. doi: 10.1093/bioinformatics/btaf248. URL https://doi.org/10.1093/bioinformatics/btaf248.

Yoo, K., Oertell, O., Lee, J., Lee, S., and Kang, J. TurboHopp: Accelerated molecule scaffold hopping with consistency models. *Advances in Neural Information Processing Systems*, 37:41157–41185, 2024.

A. Chemistry Details

A.1. Building Blocks and Reactions

The 93 selected commercial building blocks and their respective reaction centers are shown in Figure 5. For chemical reactions, we focused on cross coupling reactions to link fragments together. We chose 8 classes of robust reactions, which can be subdivided into 19 types of reaction templates, see Figure 6. We note that our reaction modelling is simplified. For example, boronic acids in building blocks ($B(OH)_2$) are replaced with boranes (BH_2); we do not consider the need for chemical protection on certain functional groups (e.g. N-Boc); we do not consider directing group effects or stoichiometry when multiple reaction centers are available. These edge cases are comparably minimal through the careful curation of building blocks to avoid such infeasible chemical reactions.

A.2. Graph Generation

Helper definitions. We annotate each building block with its reaction center atom indices $\mathcal{V}(b) \subseteq V(b)$ and its and each intrinsic atom-level graph by $H(b) \;:=\; \big(V(b),\, L(b)\big)$, where $V(b)$ is the set of atoms in b and $L(b) \subseteq V(b) \times V(b)$ is the set of covalent bonds internal to the block. Each reaction template r is annotated with a Boolean tuple $\big((l_A(r),\, l_B(r)\,\big) \in \{0,1\}^2$ describing whether reagent A or reagent B in r, respectively, contains a leaving atom.

Given the current atom graph $G_a = (V_a, L_a)$ and an atom $v \in V_a$ of degree 1, the routine $\textsc{UniqueNeighbor}(v)$ returns the *single* atom $u \in V_a$ such that $(u, v) \in L_a$. Throughout the vocabulary, every leaving-group center has exactly one neighbour.

A reaction template r is considered compatible with (b_i, v) and $(\tilde{b}, \tilde{v})$ if it queries for first and second reagent substructures that match (b_i, v) and $(\tilde{b}, \tilde{v})$, respectively.

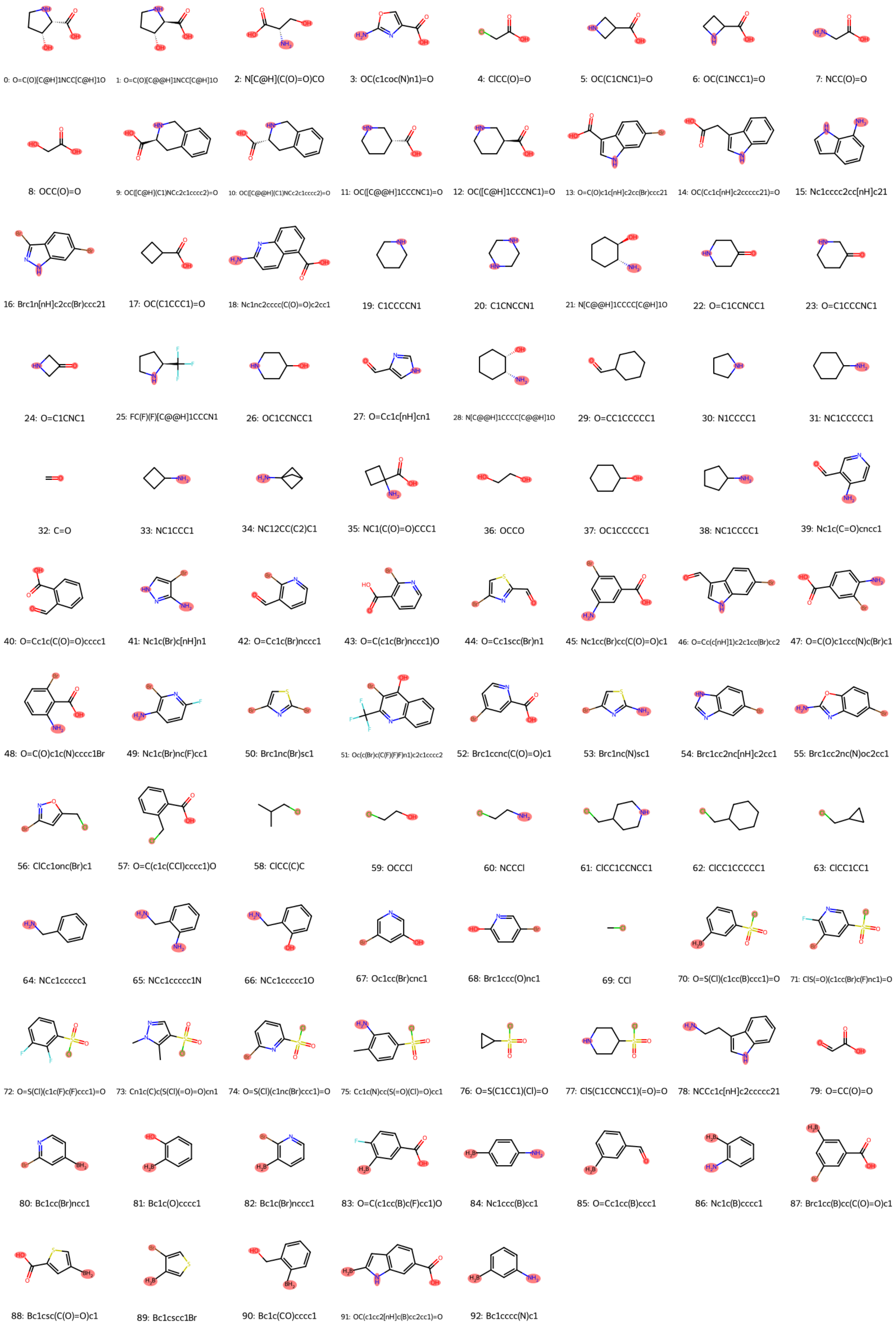

Figure 5. List of building blocks, their respective reaction centers (in red), and their SMILES representation.

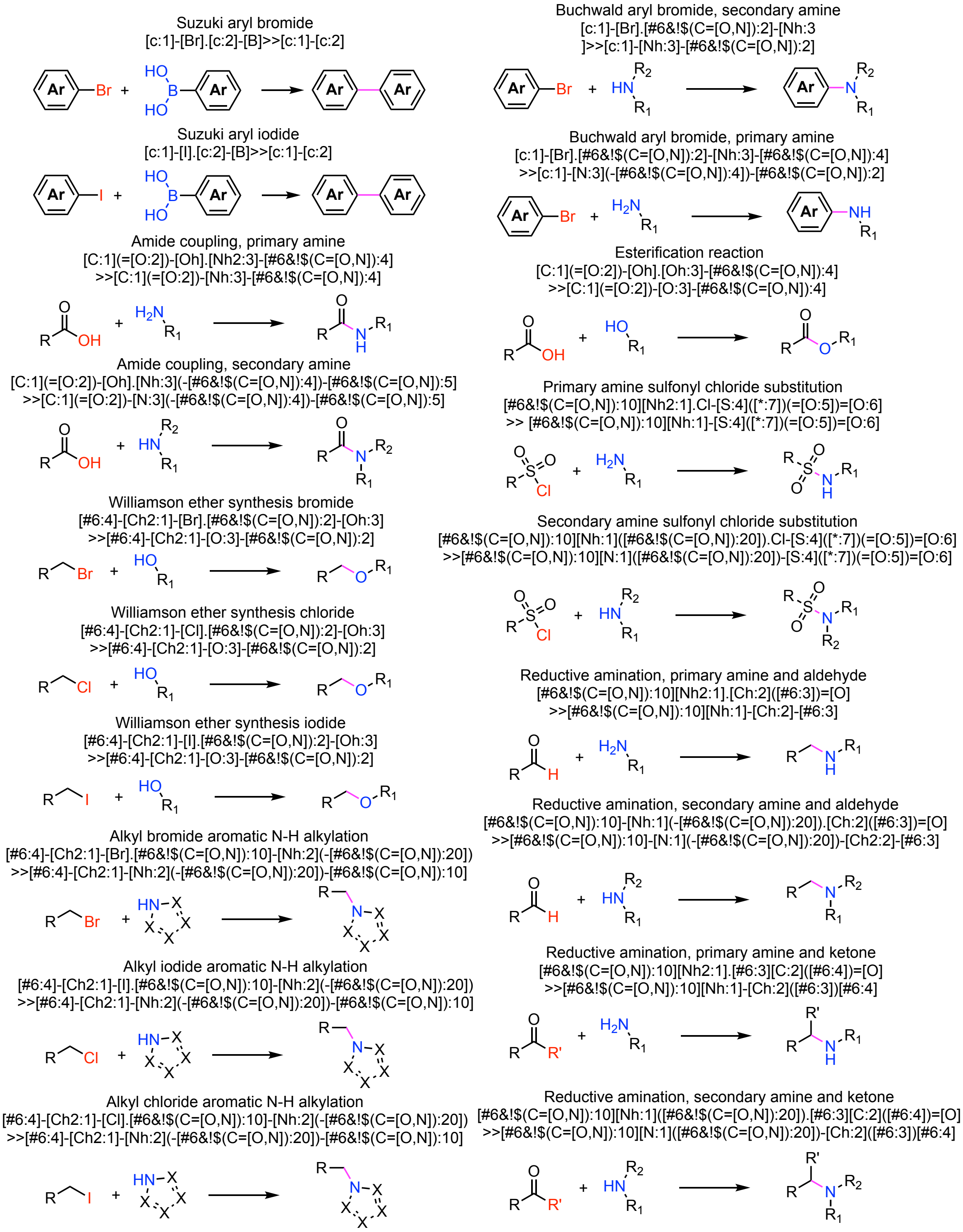
Suzuki aryl bromide
[c:1]-[Br].[c:2]-[B]>>[c:1]-[c:2]
Suzuki aryl iodide
[c:1]-[I].[c:2]-[B]>>[c:1]-[c:2]
Amide coupling, primary amine
[C:1](=[O:2])-[Oh].[Nh2:3]-[#6&!$(C=[O,N]):4]
>>[C:1](=[O:2])-[Nh:3]-[#6&!$(C=[O,N]):4]
Amide coupling, secondary amine
[C:1](=[O:2])-[Oh].[Nh:3](-[#6&!$(C=[O,N]):4])-[#6&!$(C=[O,N]):5]
>>[C:1](=[O:2])-[N:3](-[#6&!$(C=[O,N]):4])-[#6&!$(C=[O,N]):5]
Williamson ether synthesis bromide
[#6:4]-[Ch2:1]-[Br].[#6&!$(C=[O,N]):2]-[Oh:3]
>>[#6:4]-[Ch2:1]-[O:3]-[#6&!$(C=[O,N]):2]
Williamson ether synthesis chloride
[#6:4]-[Ch2:1]-[Cl].[#6&!$(C=[O,N]):2]-[Oh:3]
>>[#6:4]-[Ch2:1]-[O:3]-[#6&!$(C=[O,N]):2]
Williamson ether synthesis iodide
[#6:4]-[Ch2:1]-[I].[#6&!$(C=[O,N]):2]-[Oh:3]
>>[#6:4]-[Ch2:1]-[O:3]-[#6&!$(C=[O,N]):2]
Alkyl bromide aromatic N-H alkylation
[#6:4]-[Ch2:1]-[Br].[#6&!$(C=[O,N]):10]-[Nh:2](-[#6&!$(C=[O,N]):20])
>>[#6:4]-[Ch2:1]-[Nh:2](-[#6&!$(C=[O,N]):20])-[#6&!$(C=[O,N]):10]
Alkyl iodide aromatic N-H alkylation
[#6:4]-[Ch2:1]-[I].[#6&!$(C=[O,N]):10]-[Nh:2](-[#6&!$(C=[O,N]):20])
>>[#6:4]-[Ch2:1]-[Nh:2](-[#6&!$(C=[O,N]):20])-[#6&!$(C=[O,N]):10]
Alkyl chloride aromatic N-H alkylation
[#6:4]-[Ch2:1]-[Cl].[#6&!$(C=[O,N]):10]-[Nh:2](-[#6&!$(C=[O,N]):20])
>>[#6:4]-[Ch2:1]-[Nh:2](-[#6&!$(C=[O,N]):20])-[#6&!$(C=[O,N]):10]
Buchwald aryl bromide, secondary amine
[c:1]-[Br].[#6&!$(C=[O,N]):2]-[Nh:3
]>>[c:1]-[Nh:3]-[#6&!$(C=[O,N]):2]
Buchwald aryl bromide, primary amine
[c:1]-[Br].[#6&!$(C=[O,N]):2]-[Nh:3]-[#6&!$(C=[O,N]):4]
>>[c:1]-[N:3](-[#6&!$(C=[O,N]):4])-[#6&!$(C=[O,N]):2]
Esterification reaction
[C:1](=[O:2])-[Oh].[Oh:3]-[#6&!$(C=[O,N]):4]
>>[C:1](=[O:2])-[O:3]-[#6&!$(C=[O,N]):4]
Primary amine sulfonyl chloride substitution
[#6&!$(C=[O,N]):10][Nh2:1].Cl-[S:4]([*:7])(=[O:5])=[O:6]
>> [#6&!$(C=[O,N]):10][Nh:1]-[S:4]([*:7])(=[O:5])=[O:6]
Secondary amine sulfonyl chloride substitution
[#6&!$(C=[O,N]):10][Nh:1]([#6&!$(C=[O,N]):20]).Cl-[S:4]([*:7])(=[O:5])=[O:6]
>>[#6&!$(C=[O,N]):10][N:1]([#6&!$(C=[O,N]):20])-[S:4]([*:7])(=[O:5])=[O:6]
Reductive amination, primary amine and aldehyde
[#6&!$(C=[O,N]):10][Nh2:1].[Ch:2]([#6:3])=[O]
>>[#6&!$(C=[O,N]):10][Nh:1]-[Ch:2]-[#6:3]
Reductive amination, secondary amine and aldehyde
[#6&!$(C=[O,N]):10]-[Nh:1](-[#6&!$(C=[O,N]):20]).[Ch:2]([#6:3])=[O]
>>[#6&!$(C=[O,N]):10]-[N:1](-[#6&!$(C=[O,N]):20])-[Ch2:2]-[#6:3]
Reductive amination, primary amine and ketone
[#6&!$(C=[O,N]):10][Nh2:1].[#6:3][C:2]([#6:4])=[O]
>>[#6&!$(C=[O,N]):10][Nh:1]-[Ch:2]([#6:3])[#6:4]
Reductive amination, secondary amine and ketone
[#6&!$(C=[O,N]):10][Nh:1]([#6&!$(C=[O,N]):20]).[#6:3][C:2]([#6:4])=[O]
>>[#6&!$(C=[O,N]):10][N:1]([#6&!$(C=[O,N]):20])-[Ch:2]([#6:3])[#6:4]

Figure 6. List of chemical reactions used to connect building blocks and their SMARTS representation. Newly formed bonds are highlighted in pink.

Algorithm 3 Fragment-by-fragment assembly with COUPLE

Inputs: vocab $\mathcal{B}$, reactions $\mathcal{R}$, depth limit T

Output: atom graph G_a, building block graph $G_f = (X, E)$

```
function COUPLE($G_a,\ b_i,\ \tilde{b},\ r,\ (v_i, \tilde{v})$)
    append all atoms and bonds of $H(\tilde{b})$ to $G_a$        ▷ 1. Handle leaving groups
    if $l_A(r) = 1$ then                                          ▷ $v_i$ leaves in reagent A
        $u_i \leftarrow$ UNIQUENEIGHBOR($v_i$)
        delete atom $v_i$ (and its bond) from $G_a$
        $v_i \leftarrow u_i$                                       ▷ reroute to neighbour
    end if
    if $l_B(r) = 1$ then                                          ▷ $\tilde{v}$ leaves in reagent B
        $u_{\mathrm{t}} \leftarrow$ UNIQUENEIGHBOR($\tilde{v}$)
        delete atom $\tilde{v}$ (and its bond) from $G_a$
        $\tilde{v} \leftarrow u_{\mathrm{t}}$                       ▷ reroute to neighbour
    end if
    add covalent bond between $v_i$ and $\tilde{v}$              ▷ 2. Add the cross-bond
    return $G_a$
end function
$b_0 \leftarrow$ UniformPick($\mathcal{B}$);   $G_a \leftarrow H(b_0)$;   $G_f \leftarrow (b_0)$
for $t = 1$ to $T$ do
    $L \leftarrow$ enumerate compatible 5-tuples $\langle b_i, v, r, \tilde{b}, \tilde{v} \rangle$
    if $L = \varnothing$ then break
    end if
    $(b_i, v, r, \tilde{b}, \tilde{v}) \leftarrow$ UniformPick($L$)
    $e \leftarrow (r, v, \tilde{v})$
    $G_a \leftarrow$ COUPLE($G_a,\ b_i,\ \tilde{b},\ r, (v, \tilde{v})$)
    $G_f \leftarrow G_f \cup \big(b_i \xrightarrow{e} \tilde{b}\big)$
end for
return $(G_a, G_f)$
```

B. Method Details

B.1. Building Block-Level Representations

Let $X \in \{0,1\}^{N\times|\mathcal{B}|+1}$ be a one-hot matrix where the i^{th} row encodes the identity of the i^{th} building block, and let $E \in \{0,1\}^{N\times N\times|\mathcal{R}|V^2_{max}+2}$, where $V_{max} = \max_b |\mathcal{V}(b)|$. A non-zero entry $E_{ijr(v_i,v_j)} = 1$ signals that block i (center v_i) couples to block j (center v_j) via reaction r. Graphs (X, E) belonging to molecules containing $n < N$ building blocks are padded to N.

Reserved Channels. We reserve a dedicated *masked* (absorbing) token in both vocabularies:

$$\pi_X \in \{0,1\}^{|\mathcal{B}|}, \qquad \pi_E \in \{0,1\}^{|\mathcal{R}|\,V^2_{\max}}, \tag{8}$$

where π_X (resp. π_E) is the one-hot vector whose single 1-entry corresponds to the masked node (resp. edge) channel. Besides the masked channel we keep a dedicated *no-edge* channel, encoded by the one-hot vector

$$\lambda_E \in \{0,1\}^{|\mathcal{R}|\,V^2_{\max}}, \tag{9}$$

so every edge slot may take one of three mutually exclusive states: a concrete coupling label, the no-edge token λ_E, or the masked token π_E.

B.2. Atom-Level Representations

The SEMLAFLOW (Irwin et al., 2025) architecture propagates and updates invariant and equivariant features at the atom level. To ensure consistency with this framework, we calculate for each input graph (X_t, E_t) atom-level one-hot atom and bond features. Crucially, these features must be flexible to arbitrary masking present in X_t and E_t. With this in mind we set each atom feature $X_t^{atom}[i, a]$ to a concatenation of one-hot encodings

$$X_t^{atom}[i,a] \;=\; \Big(\underbrace{\delta_{\mathrm{sym}(i,a)}}_{\text{9-way one-hot}},\ \mathbb{1}[\mathrm{ring}(i,a)],\ \mathbb{1}[a \in \mathcal{V}(X_i)]\Big) \in \{0,1\}^{\,9+2}, \tag{10}$$

where $\delta_{\mathrm{sym}}(i,a)$ is the onehot vector over possible atom types (C, N, O, B, F, Cl, Br, S, `[MASK]`) and ring(i,a) denotes whether or not the atom is a member of a ring. Similarly, we calculate the a bond feature matrix

$$E_t^{\text{atom}}[a_i, a_j] = \begin{cases} \delta_{\text{order}}(a_i, a_j), & \text{bond is present,} \\ \mathbf{0}_5, & \text{otherwise.} \end{cases} \tag{11}$$

where $\delta_{\text{order}}(a_i, a_j)$ is the onehot tensor over possible bond orders (single, double, triple, aromatic, `[MASK]`) between a_i and a_j. E_t^{atom} is populated by loading the known bonds and respective bond orders within denoised building blocks. If some building block X_i is noised, all edges between its constituent atoms $E_t^{atom}[i : i + M, i : i + M]$ are set to the masked one-hot index. For graphs (X_t, E_t) corresponding to valid molecules in which all nodes and edges are denoised, we simply obtain the full bond feature matrix from the molecule described by (X_t, E_t).

B.3. Training Algorithm

Algorithm 4 Training step for SYNCOGEN

```
$t \sim \mathcal{U}(0,1)$
$(X_t, E_t) \leftarrow q_t(X_0, E_0)$
$C_1 \sim \mathcal{N}(0, I)$
$(\tilde{C}_0, \tilde{C}_t) \leftarrow \textsc{Pair}(C_0, S_0, C_1, t, X_t)$          ▷ center and interpolate coordinates (Sec. ??)
$(L_t^X, L_t^E, \hat{\tilde{C}}_0^t) \leftarrow f_\theta(X_t, E_t, \tilde{C}_t, n, t)$
$\mathcal{L} \leftarrow \mathcal{L}_{\text{graph}} + \mathcal{L}_{\text{MSE}} + \mathcal{L}_{\text{pair}}$          ▷ total loss (Sec. B.11)
$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$
```

B.4. Sampling Algorithm

Algorithm 5 Sampling procedure for SYNCOGEN

```
$n \sim \text{Cat}(\pi_{\text{frag}});\ (X_1, E_1) \leftarrow (\pi_X, \pi_E);\ S_1[i,a] \leftarrow \mathbf{1}[i < n]$          ▷ draw $n$, initialize masks
$C_1 \sim \mathcal{N}(0, I);\ \tilde{C}_1 \leftarrow C_1 - \bar{C}_{1,S_1}$          ▷ center Gaussian prior by initial mask
**for** $t = 1$ **down to** $0$ **do**
    $\tilde{C}_t \leftarrow C_t - \bar{C}_{t,S_t}$;
    $(L_t^X, L_t^E, \hat{\tilde{C}}_0^t) \leftarrow f_\theta(X_t, E_t, \tilde{C}_t, n, t)$
    $\tilde{L}_t^E \leftarrow \textsc{SampleEdges}(L_t^E, n)$          ▷ enforce one parent per building block (Sec. ??)
    $X_{t-\Delta t} \leftarrow \textsc{CatSample}(L_t^X);\ E_{t-\Delta t} \leftarrow \textsc{CatSample}(\tilde{L}_t^E)$          ▷ take reverse step (App. B.6)
    $C_{t-\Delta t} \leftarrow C_t + \Delta t(\hat{\tilde{C}}_0^t - \tilde{C}_t)$
    $(X_t, E_t, C_t, S_t) \leftarrow (X_{t-\Delta t}, E_{t-\Delta t}, C_{t-\Delta t}, S_{t-\Delta t})$
**end for**
$(L^X, L^E, \hat{\tilde{C}}_0) \leftarrow f_\theta(X_0, E_0, \tilde{C}_0, n, 0)$          ▷ final deterministic denoise ($t = 0$)
$\hat{X}_0 \leftarrow \arg\max_k L_\theta^X[\cdots, k];\ \hat{E}_0 \leftarrow \arg\max_k L_\theta^E[\cdots, k];\ \hat{C}_0 \leftarrow \hat{\tilde{C}}_0 - \bar{\hat{\tilde{C}}}_{0,S_0}$
**return** $(\hat{X}_0, \hat{E}_0, \hat{C}_0)$
```

B.5. Building Block Logit Predictions

The SEMLAFLOW(Irwin et al., 2025) backbone outputs atom–atom edge features $E_\theta^{\text{atom}} \in \mathbb{R}^{B\times(NM)\times(NM)\times d_{\text{edge}}}$. To obtain building block-level tensors we apply two parallel 2-D convolutions (one for nodes, one for edges) with stride M, followed by MLP classifiers that map the pooled features back to their original one-hot vocabularies.

Stride-pooled convolution. Let d_{edge} be the latent edge feature dimension. Each stream uses the block

$$\text{Conv2d}(d_{\text{edge}} \to d_{\text{edge}},\, k = M,\, s = M) \xrightarrow{\text{SiLU}} \text{Conv2d}(d_{\text{edge}} \to d_{\text{edge}},\, k = 1,\, s = 1), \tag{12}$$

so every $M \times M$ atom patch collapses to a single building block entry. This produces

$$X_{\text{pool}} \in \mathbb{R}^{B\times d_{\text{edge}}\times N}, \qquad E_{\text{pool}} \in \mathbb{R}^{B\times d_{\text{edge}}\times N\times N}. \tag{13}$$

Node head. We flatten X_{pool} along its channel axis, concatenate the residual building block one-hot matrix X_t, and pass the result through a two-layer MLP to obtain

$$L_\theta^{X_t} \in \mathbb{R}^{B\times N\times|\mathcal{B}|}. \tag{14}$$

Edge head. We concatenate E_{pool} with the residual building block-edge one-hot tensor E_t, apply an analogous two-layer MLP, and symmetrise to produce

$$L_\theta^{E_t} \in \mathbb{R}^{B\times N\times N\times|\mathcal{R}|V_{\max}^2}. \tag{15}$$

Atom Features. The SEMLAFLOW(Irwin et al., 2025) backbone additionally outputs atom-level node features $X_\theta^{\text{atom}} \in \mathbb{R}^{B\times(NM)\times d_{\text{node}}}$, which are incorporated into E_θ^{atom} via a bond refinement message-passing layer. We find that extracting both building block and edge logits directly from the refined features E_θ^{atom} marginally improves performance relative to separately predicting $L_\theta^{X_t}$ from X_θ^{atom} and $L_\theta^{E_t}$ from E_θ^{atom}.

B.6. Discrete Noising Scheme

Following (Sahoo et al., 2024), we adopt an absorbing (masked) state noising scheme for X_0 and E_0:

$$q(X_t \mid X_0) = \text{Cat}\big(X_t;\, \alpha_t X_0 + (1-\alpha_t)\pi_X\big), \qquad q(E_t \mid E_0) = \text{Cat}\big(E_t;\, \alpha_t E_0 + (1-\alpha_t)\pi_E\big). \tag{16}$$

where $(\alpha_t)_{t\in[0,1]}$ is the monotonically decreasing noise schedule introduced in Sec. **??**.

Reverse categorical posterior. For node identities we have

$$q(X_s \mid X_t,\, X_0) = \begin{cases} \text{Cat}(X_s; X_t), & X_t \neq \pi_X, \\ \text{Cat}\big(X_s; \dfrac{(1-\alpha_s)\pi_X + \alpha_s\, X_\theta^t}{1-\alpha_t}\big), & X_t = \pi_X, \end{cases} \tag{17}$$

and, analogously, for edge labels

$$q(E_s \mid E_t,\, E_0) = \begin{cases} \text{Cat}(E_s; E_t), & E_t \neq \pi_E \\ \text{Cat}\big(E_s; \dfrac{(1-\alpha_s)\pi_E + \alpha_s\, E_\theta^t}{1-\alpha_t}\big), & E_t = \pi_E, \end{cases} \tag{18}$$

where $s < t$. Equations 17 and 18 are the direct translation of the reverse noising process described by (Sahoo et al., 2024) into SYNCOGEN's node–edge representation.

B.7. Noise Schedule Parameterization

Following MDLM (Sahoo et al., 2024), we parameterize the discrete noising schedule via $\alpha_t = e^{-\sigma(t)}$, where $\sigma(t) : [0,1] \to \mathbb{R}^+$. In all experiments, we adopt the **linear schedule**:

$$\sigma(t) = \sigma_{\max} t, \tag{19}$$

where $\sigma_{\max}$ is a large constant; we use $\sigma_{\max} = 10^8$ as in the original MDLM setup.

Edge Symmetrization. After drawing the upper-triangle entries of the one-hot edge tensor E_s in either the forward or reverse noising process, we enforce symmetry by copying them to the lower triangle:

$$E_{s,jie} \;=\; E_{s,ije}, \qquad 0 \le i < j < n,\;\; e \in \mathcal{R}V^2_{\max}.$$

B.8. Positional Embeddings

Though SemlaFlow(Irwin et al., 2025) is permutationally invariant by design with respect to atom positions, SynCoGen dataset molecules require that atom order be fixed and grouped by building block for reconstruction purposes. To enforce this during training, we intentionally break permutation invariance by generating and concatenating to each input coordinate sinusoidal positional embeddings representing both global atom index and building block index.

B.9. Hyperparameters

We train SynCoGen for 50 epochs with a batch size of 128 and a global batch size of 512. All models are trained with a linear noise schedule (see Appendix B.7), with the SUBS parameterization enabled. During training, a random conformer for each molecule is selected, then centered and randomly rotated to serve as the ground-truth coordinates C_0. All atomic coordinates are normalized by a constant Z_c describing the standard deviation across all training examples. For the pairwise distance loss $\mathcal{L}_{pair}$ we set d to 3Å, adjusted for normalization. During training, for each recentered input-prior pair $(\tilde{C}_1, \tilde{C}_0)$ we rotationally align C_1 to C_0. When training with noise scaling and the bond loss time threshold, we set the noise scaling coefficient to 0.2 , and the time threshold to 0.25, above which bond length losses are zeroed. When training with auxiliary losses we set the weights for the pairwise, sLDDT, and bond length loss components to 0.4, 0.4, and 0.2, respectively. Anonymized code can be found at `https://anonymous.4open.science/r/SynCoGen-13F7`.

B.10. Computational Resources Used

We train all models on 2 H100-80GB GPUs.

B.11. Training Losses

Here we define several loss terms that have proved useful for stabilising training on 3-D geometry.

By default, SynCoGen is trained with $\mathcal{L}_{\text{MSE}}$ and L_{pair} as coordinate losses.

For a prediction $\big(L_\theta^{X_t},\, L_\theta^{E_t},\, \hat{\tilde{C}}_0^{\,t}\big) = f_\theta\big(X_t, E_t, \tilde{C}_t, n, t\big),\; X_\theta^t = \mathrm{softmax}\big(L_\theta^{X_t}\big),\; E_\theta^t = \mathrm{softmax}\big(L_\theta^{E_t}\big)$:

Graph loss. Let X_0 and E_0 be the clean node and edge tensors. Following the MDLM implementation (Sahoo et al., 2024), we weigh the negative log-likelihood at step t by

$$w_t = \frac{\Delta\sigma_t}{\exp(\sigma_t) - 1}, \qquad \Delta\sigma_t = \sigma_t - \sigma_{t-1},\;\; \sigma_0 = 0, \tag{20}$$

where σ_t is the discrete noise level. The discrete (categorical) loss is then

$$\mathcal{L}_{\text{graph}} = w_t\big(-\log X_\theta^t[X_0] \; -\log E_\theta^t[E_0]\big), \tag{21}$$

i.e. the cross-entropy between the one-hot ground truth and the predicted distributions for both nodes and edges.

MSE loss. Let $S_0 \in \{0,1\}^{N\times M}$ mask the atoms that exist in the clean structure and C_t be the noisy coordinates. Denote $\mathcal{A}_{S_0} = \big\{(i,a) : S_0[i,a] = 1\big\}$.

$$\mathcal{L}_{\text{MSE}} = \frac{1}{|\mathcal{A}_{S_0}|} \sum_{(i,a)\in\mathcal{A}_{S_0}} \big\|\hat{C}_0[i,a] - C_0[i,a]\big\|_2^2, \tag{22}$$

Pairwise loss.

$$\mathcal{L}_{\text{pair}} = \sum_{\substack{(i,a)<(j,b) \\ \|C_0[i,a]-C_0[j,b]\|_2 \le d}} S_0[i,a]\, S_0[j,b]\, \big(\|\hat{C}_0[i,a] - \hat{C}_0[j,b]\|_2 - \|C_0[i,a] - C_0[j,b]\|_2\big)^2, \tag{23}$$

where d is the distance cut-off for pairwise terms. The default total loss value for the model is therefore

$$\mathcal{L}_{\text{SynCoGen}} = \mathcal{L}_{\text{graph}} + \mathcal{L}_{\text{MSE}} + \mathcal{L}_{\text{pair}}. \tag{24}$$

Smooth-LDDT loss (Abramson et al., 2024). Let $d_{ij}^{0} := \|C_0[i] - C_0[j]\|_2$ and $d_{ij}^{\text{pred}} := \|\hat{C}_0[i] - \hat{C}_0[j]\|_2$ be ground-truth and predicted inter-atomic distances, respectively. For each pair of atoms within a 15 Å cutoff in the reference structure we compute the per-pair score

$$\text{sLDDT}_{ij} = \frac{1}{4}\sum_{k=1}^{4} \sigma\big(\tau_k - |d_{ij}^{\text{pred}} - d_{ij}^{0}|\big), \quad \big[\tau_1, \tau_2, \tau_3, \tau_4\big] = [0.5, 1, 2, 4]\ \text{Å},$$

where $\sigma(x) = 1/(1+e^{-x})$ is the logistic function. The smooth-LDDT loss averages $1 - \text{sLDDT}_{ij}$ over all valid pairs,

$$\mathcal{L}_{\text{sLDDT}} = \frac{\sum_{i<j} \mathbb{1}[d_{ij}^{0} < 15]\, S_0[i]\, S_0[j]\, \big(1 - \text{sLDDT}_{ij}\big)}{\sum_{i<j} \mathbb{1}[d_{ij}^{0} < 15]\, S_0[i]\, S_0[j]}. \tag{25}$$

Bond-length loss. Given a set of intra-fragment bonds $\text{bonds} = \{(p, q)\}$ extracted from the vocabulary, we penalise deviations in predicted bond lengths:

$$\mathcal{L}_{\text{bond}} = \frac{1}{|\text{bonds}|} \sum_{(p,q)\in\text{bonds}} \big|\|\hat{C}_0[p] - \hat{C}_0[q]\|_2 - \|C_0[p] - C_0[q]\|_2\big|. \tag{26}$$

Self-Conditioning. The modified SemlaFlow (Irwin et al., 2025) backbone operates on node and edges features at the atomic level, but outputs unnormalized prediction logits $\hat{X}_0 \in \{0,1\}^{N\times|\mathcal{B}|}$ and $\hat{E}_0 \in \{0,1\}^{N\times N\times|\mathcal{R}|V_{max}^2}$. We therefore implement modified self-conditioning for SynCoGen that projects previous step graph predictions $\hat{X}_{0cond}$ and $\hat{E}_{0cond}$ to the shape of X_t^{atom} and E_t^{atom} using an MLP.

B.12. Conformer generation

We randomly assembled 50 molecules with the reaction graph, and used the standard conformational search (iMTD-GC) in CREST with GFN-FF to find all reference conformers. For both SynCoGen and RDKit ETKDG, we sampled 50 conformers per molecule and computed the coverage and matching scores. We used a relatively strict RMSD threshold of $\tau = 0.75$.

Formally, COV is defined as:

$$\text{COV} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\Big[\min_{1\le j\le M} \text{RMSD}(m_i,\, g_j) \le \tau\Big], \tag{27}$$

where $\mathbf{1}[\cdot]$ is the indicator function, m_i are the N generated conformers and g_j are the M reference conformers. And MAT is defined as:

$$\text{MAT} = \frac{1}{N}\sum_{i=1}^{N} \min_{1\le j\le M} \text{RMSD}(m_i,\, g_j). \tag{28}$$

B.13. Molecular inpainting

For the inpainting experiments in Section 5.2, we keep two fragments $\mathcal{D} = \{\mathcal{D}^{(1)}, \mathcal{D}^{(2)}\}$ and their coordinates fixed and sample the remaining part of the molecule. We follow Algorithm B.4 and initialize the graph prior X_1 with the one-hot encoding of the desired fragment i at a specified node index in the graph (decided at random or based on the structure of the original molecule, so that it matches its scaffold). For each denoised fragment $\mathcal{D}^{(i)}$, we replace its coordinates at each time $t > 0.03$ during sampling by

$$C_t^{(i)} \;=\; (1-t)\,\tilde{C}_0^{(i)} \;+\; t\,\tilde{C}_1^{(i)},$$

where $\tilde{C}_0^{(i)}$ and $\tilde{C}_1^{(i)}$ are the centered ground-truth and prior coordinates of fragment i, respectively, and all other fragments are updated as shown in Algorithm B.4. For any $t \leq 0.03$, which for 100 sampling steps amounts for the last three steps in the path, we follow normal Euler steps as shown in Algorithm B.4 to allow a refinement of the fixed coordinates in line with the rest of the predicted ones for the rest of the fragments. We empirically observed that this led to molecules with lower average energies.

C. Baseline comparisons.

For all baselines we sampled 1000 molecules with random seeds on an A100 GPU and reported averaged results over three runs.

SemlaFlow We evaluated SemlaFlow using the sampling script in the official codebase on GitHub[1]. We reported results for a model trained on the GEOM(Axelrod & Gomez-Bombarelli, 2022) dataset (by sampling from the checkpoints provided in the repository) and from a model trained on our dataset (see Table 1). We trained SemlaFlow using the default hyperparameters for 150 epochs on a single conformer per molecule.

EQGAT-diff, MiDi, JODO, FlowMol We evaluated EQGAT-diff, Midi, JODO, using their official implementations provided on GitHub[2]. We modified the example sampling script to save molecules as outputted from the reverse sampling, without any post-processing. For MiDi, we evaluated the uniform model. For FlowMol, both CTMC and Gaussian models were evaluated and reported.

D. Extended results and discussion

D.1. Sampling Ablations

By default, SynCoGen implements a linear noise schedule and samples for 100 timesteps. To evaluate the effect of step count and noise schedule choice on performance, we provide experiments with step count decreased to 50 and 20, as well as modified noising to follow a loglinear and geometric schedule. All results listed subsequently can be assumed to use the default noise schedule and step count.

We additionally follow FoldFlow to implement *inference annealing*, a time-dependent scaling on Euler step size that was found to empirically improve designability results (Bose et al., 2024). In our experiments, we multiply the Euler step size at time t by $5t$, $10t$, and $50t$.

We find that noising and de-noising building blocks according to a linear noise schedule generally achieves the best performance, which during inference sees most unmasking occur in the final steps. An aggressive denoising schedule for the discrete fragments yield in significantly worse validity (Geometric and Loglinear). Nevertheless, inference annealing that marginally speeds up discrete denoising can slightly improve discrete generation validity. As a sanity check to evaluate whether simultaneous generation is still necessary for good performance using SynCoGen, we evaluate an inference configurations where all building blocks are noised until a single final prediction step (FinalOnly)where we find performance using the default parameters to be superior.

[1] `https://github.com/rssrwn/semla-flow/`, available under the MIT License

[2] `https://github.com/jule-c/eqgat_diff/`, `https://github.com/cvignac/MiDi`, `https://github.com/GRAPH-0/JODO`, `https://github.com/Dunni3/FlowMol`, available under the MIT License

Table 3. Sampling ablations. Results are averaged over 1000 generated samples except retrosynthesis solve rate (out of 100). All results shown are at 50 epoches rather than 100 epoches in Table 1.

	Primary metrics					Secondary metrics		
Method	Valid. ↑	AiZyn. ↑	Synth. ↑	GFN-FF ↓	xTB-2 ↓	PB Val. ↑	Div. ↑	Nov. ↑
Linear-100 (Default)	93.5	55	70	4.933	-0.92	78.3	0.79	94.1
Linear-20	82.4	56	68	5.102	-0.91	71.3	0.78	94.9
Linear-50	92.0	50	65	4.890	-0.91	78.9	0.78	93.6
Geometric-100	48.2	61	68	5.206	-0.84	72.0	0.80	91.7
Loglinear-100	60.3	56	64	5.182	-0.87	70.1	0.80	91.7
Annealing-5t	94.7	52	58	5.001	-0.93	79.1	0.78	94.1
Annealing-10t	93.5	42	68	4.870	-0.91	82.8	0.78	94.2
Annealing-50t	85.1	51	64	4.972	-0.82	86.7	0.76	94.6
FinalOnly	69.7	39	68	5.260	-0.92	70.1	0.76	94.1

D.2. Metrics

We here describe metric computation details that are absent in the main text.

For synthesizability evaluation, we used the public AiZynthFinder and Syntheseus models. Due to the speed of these models, we only evaluate 100 randomly sampled generated examples. For AiZynthFinder, we used the USPTO policy, the Zinc stock, and we extended the search time to 800 seconds with an iteration limit of 200 seconds. For Syntheseus, we used the LocalRetro model with Retro* search under default settings, with Enamine REAL strict fragments as the stock. We additionally appended our building blocks as the stock but found no meaningful difference in solved rates, presumably as most of our building blocks are already in the utilized stock. We note that we replaced all boranes with boronic acids due to simplifications made in our modelling (see Appendix A.2).

For energy evaluation, all results are from single-point calculations. For GFN-FF, we report the total energy minus the bond energies (equivalent to the sum of angle, dihedral, bond repulsion, electrostatic, dispersion, hydrogen bond, and halogen bond energies) as the intramolecular non-bond energies, and average it over number of atoms. For GFN2-xTB, we report the dispersion interaction energies as the intramolecular non-covalent energies. We note that the total energies and bonded energies follow very similar trends. We note that MMFF94 energies are not parameterized for boron; therefore, we report them only for the Wasserstein distances in Appendix D.3 and inpainting task in Table 6. Figures 3 and 7 show distributions obtained from 1,000 molecules generated by each generative method, along with 50,000 subsampled molecules from their respective training datasets. Gaussian kernel density estimation (bandwidth = 0.15) was used for linear distributions, while von Mises kernel density estimation ($\kappa = 25$) was applied for circular distributions. Wasserstein-1 distances (computed linearly for lengths and energies, and on the circle for angles and dihedrals) were calculated using the Python Optimal Transport Package (Flamary et al., 2021).

D.3. *De novo* 3D molecule generation

Additional sample molecules and their synthetic pathways can be found in Figures 8 and 10. We further took a subset of random molecules and performed GFN2-xTB geometry optimization, and find that our generated samples closely recapitulates the optimized geometries (Figure 9).

Table 4. Comparison of mean coverage (COV) and matching accuracy (MAT) for RDKit ETKDG and zero-shot generation using SYNCOGEN.

Method	**COV (%) ↑**	**MAT (Å) ↓**
RDKit	0.692	0.657
SYNCOGEN	0.614	0.693

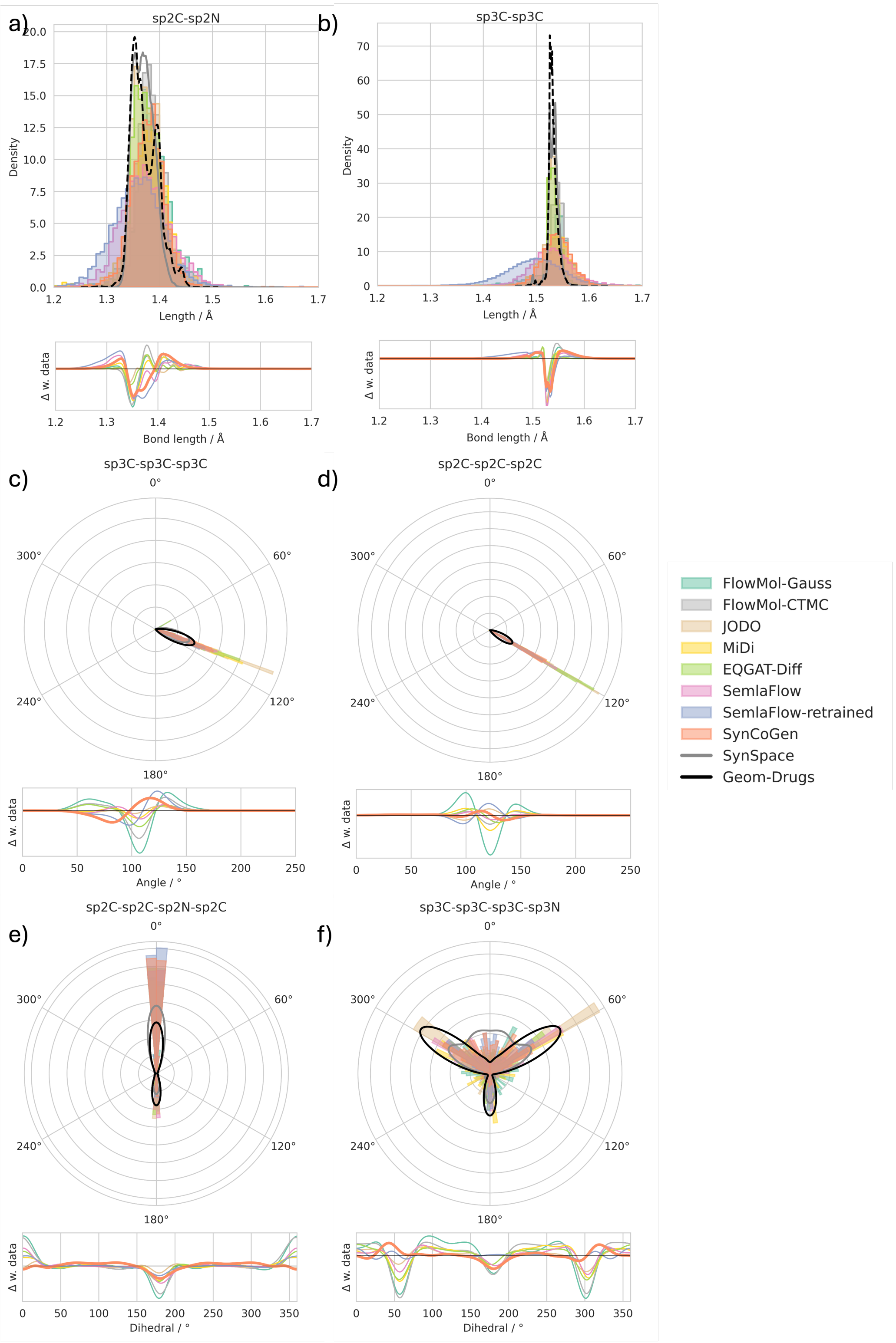

Figure 7. **Additional conformer bond length, angle, ane dihedral distribution comparisons.** a-b) Bond lengths, c-d) bond angles, e-f) dihedral angles. Comparison to baseline generative models and the kernel density estimation of the training data are shown as differences or as solid lines.

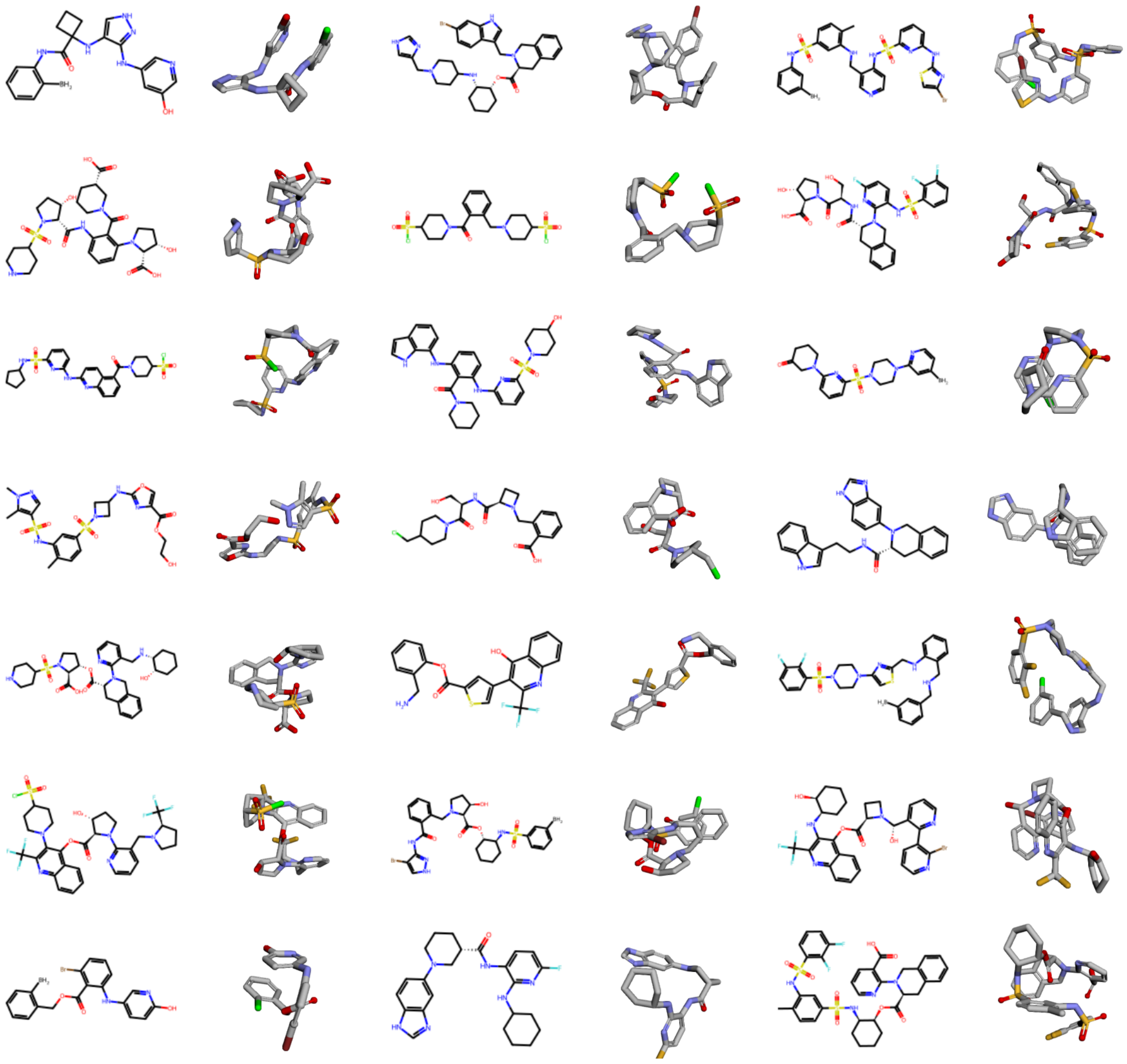

Figure 8. Additional randomly sampled molecules from SynCoGen.

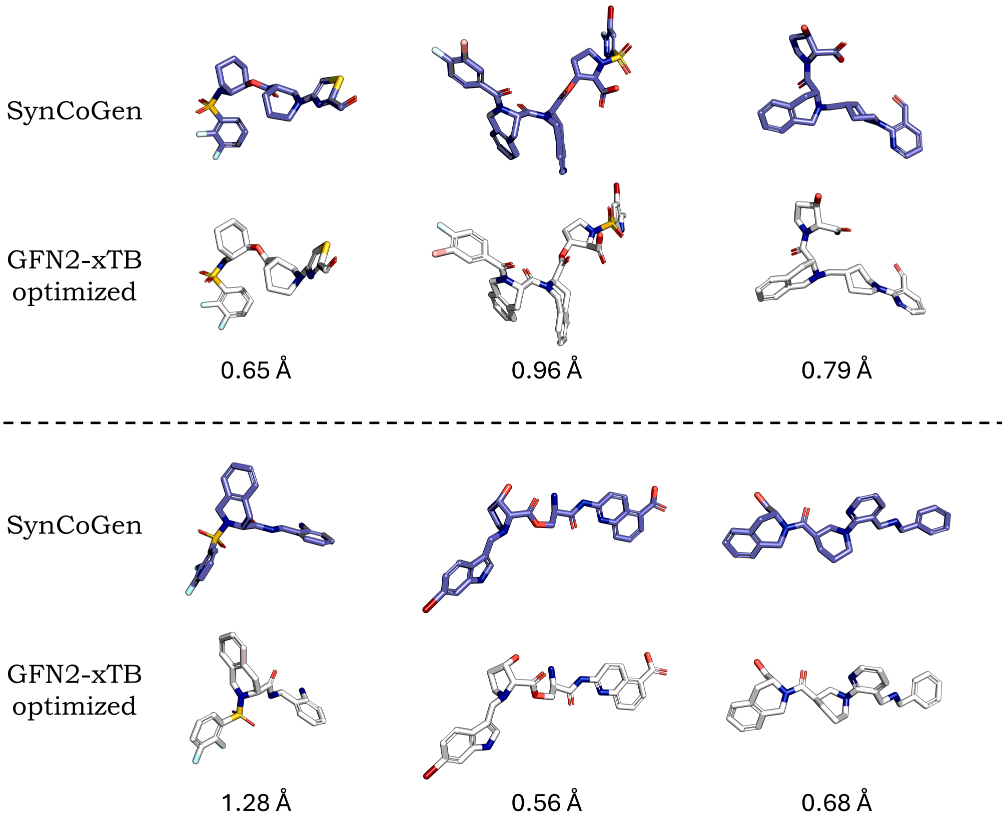

Figure 9. A subset of randomly sampled molecules from SynCoGen and further optimized by GFN2-xTB until convergence. Alignment RMSD is shown below the molecular structures.

Table 5. Wasserstein-1 distance (W_1) and Jensen–Shannon divergence (JSD) for the generative models (lower is better). For bond lengths, angles, and dihedrals, we computed the average W_1 and JSD for the top 10 prevalent lengths/angles/dihedrals. Comparisons are made to the respective training set.

(a) Bond dihedrals

Method	W_1	JSD
SynCoGen	7.01	0.29
SemlaFlow-retrained	6.50	0.22
SemlaFlow	7.76	0.28
EQGAT-Diff	8.48	0.29
MiDi	9.32	0.38
JODO	5.47	0.31
FlowMol-CTMC	13.69	0.35
FlowMol-Gauss	18.85	0.46

(b) Bond angles

Method	W_1	JSD
SynCoGen	1.36	0.22
SemlaFlow-retrained	1.64	0.28
SemlaFlow	1.18	0.21
EQGAT-Diff	1.37	0.16
MiDi	1.41	0.21
JODO	0.59	0.12
FlowMol-CTMC	1.90	0.24
FlowMol-Gauss	3.68	0.30

(c) Bond lengths

Method	W_1	JSD
SynCoGen	0.0171	0.34
SemlaFlow-retrained	0.0320	0.48
SemlaFlow	0.0200	0.38
EQGAT-Diff	0.0039	0.13
MiDi	0.0142	0.31
JODO	0.0034	0.12
FlowMol-CTMC	0.0089	0.20
FlowMol-Gauss	0.0152	0.28

(d) `xtb-2` non-covalent E

Method	W_1	JSD
SynCoGen	0.0838	0.33
SemlaFlow-retrained	0.0125	0.16
SemlaFlow	0.0249	0.16
EQGAT-Diff	0.0073	0.12
MiDi	0.0084	0.14
JODO	0.0031	0.11
FlowMol-CTMC	0.0605	0.26
FlowMol-Gauss	0.0322	0.19

(e) `GFN-FF` non-bonded E

Method	W_1	JSD
SynCoGen	1.37	0.28
SemlaFlow-retrained	1.09	0.22
SemlaFlow	1.52	0.16
EQGAT-Diff	1.69	0.18
MiDi	1.80	0.19
JODO	1.33	0.12
FlowMol-CTMC	1.53	0.17
FlowMol-Gauss	2.13	0.17

(f) `MMFF` total E

Method	W_1	JSD
SynCoGen	6.59	0.089
SemlaFlow-retrained	54.63	0.22
SemlaFlow	69.56	0.24
EQGAT-Diff	4.80	0.076
MiDi	19.00	0.11
JODO	22.07	0.11
FlowMol-CTMC	41.95	0.15
FlowMol-Gauss	26.96	0.14

D.4. Molecular inpainting experiments

Three protein–ligand complexes (PDB IDs 7N7X[3], 5L2S[4] and 3NDX[5]) were selected for molecular inpainting of the ligand structures. Note that for 3NDX, we substituted the protein file with that from PDB entry 4EYR for docking, because of the corrupted 3NDX PDB protein file – nontheless both entries contain the same protease and ligand, just in a different conformation. These ligands were chosen because they are prominent FDA-approved drugs, and that they are typically challenging to synthesize but the key functional groups are present in our building blocks. Specifically, 3NDX contains ritonavir, a prominent HIV protease inhibitor on World Health Organization's List of Essential Medicines; 5L2S contains abemaciclib, a anti-cancer kinase inhibitor that is amongst the largest selling small molecule drug; 7N7X contains berotralstat, a recently approved drug that prevents hereditary angioedema.

In addition to the experiments in Section 5.2, we evaluate our model's conditional sampling performance – within the fragment linking framework – against the state-of-the-art model DiffLinker (Igashov et al., 2024). We emphasize that while DiffLinker is trained for fragment linking, our model performs zero-shot fragment linking without any finetuning. For both models the size of the linker was chosen so that it matches that of the original ligand: 2 extra nodes were sampled for SynCoGen and 15 linking atoms for DiffLinker in the case of 5L2S, while 3 extra nodes and 25 linking atoms were sampled for 3NDX and 7N7X. We specified leaving groups (for SynCoGen) and anchor points (for DiffLinker) so that the fragments are linked at the same positions as in the ligand. Results are shown in Table 6. We note that no retrosynthetic pathways were found for the molecules in DiffLinkers, while SynCoGen models synthetic pathways and synthetic pathways can be easily drawn, with examples for 3NDX shown in Figure 10. Additionally SynCoGen achieves 100% connectivity as it uses reaction-based assembly, whereas DiffLinker can sample disconnected fragments.

[3] `https://www.rcsb.org/structure/7N7X`

[4] `https://www.rcsb.org/structure/5L2S`

[5] `https://www.rcsb.org/structure/3NDX`

Table 6. Molecular inpainting task. Results are averaged over 1000 generated samples except retrosynthesis solve rate (out of 100).

Method	Target	AiZyn. ↑	Synth. ↑	Valid. ↑	Connect. ↑	MMFF ↓	GFN-FF ↓	xTB-2 ↓	Diversity ↑	PB Val. ↑
DiffLinker	5L2S	0	0	95.8	95.09	14.22	7.52	-0.95	0.60	49.3
	3NDX	0	0	93.7	81.86	20.01	8.49	-1.03	0.81	35.0
	7N7X	0	0	95.8	74.65	20.51	7.99	-1.09	0.78	37.5
SYNCOGEN	5L2S	73	79	57.6	100	10.11	6.77	-0.78	0.62	27.3
	3NDX	72	58	46.9	100	12.80	6.58	-0.86	0.64	32.0
	7N7X	53	69	50.6	100	4.243	6.60	-0.80	0.67	56.1

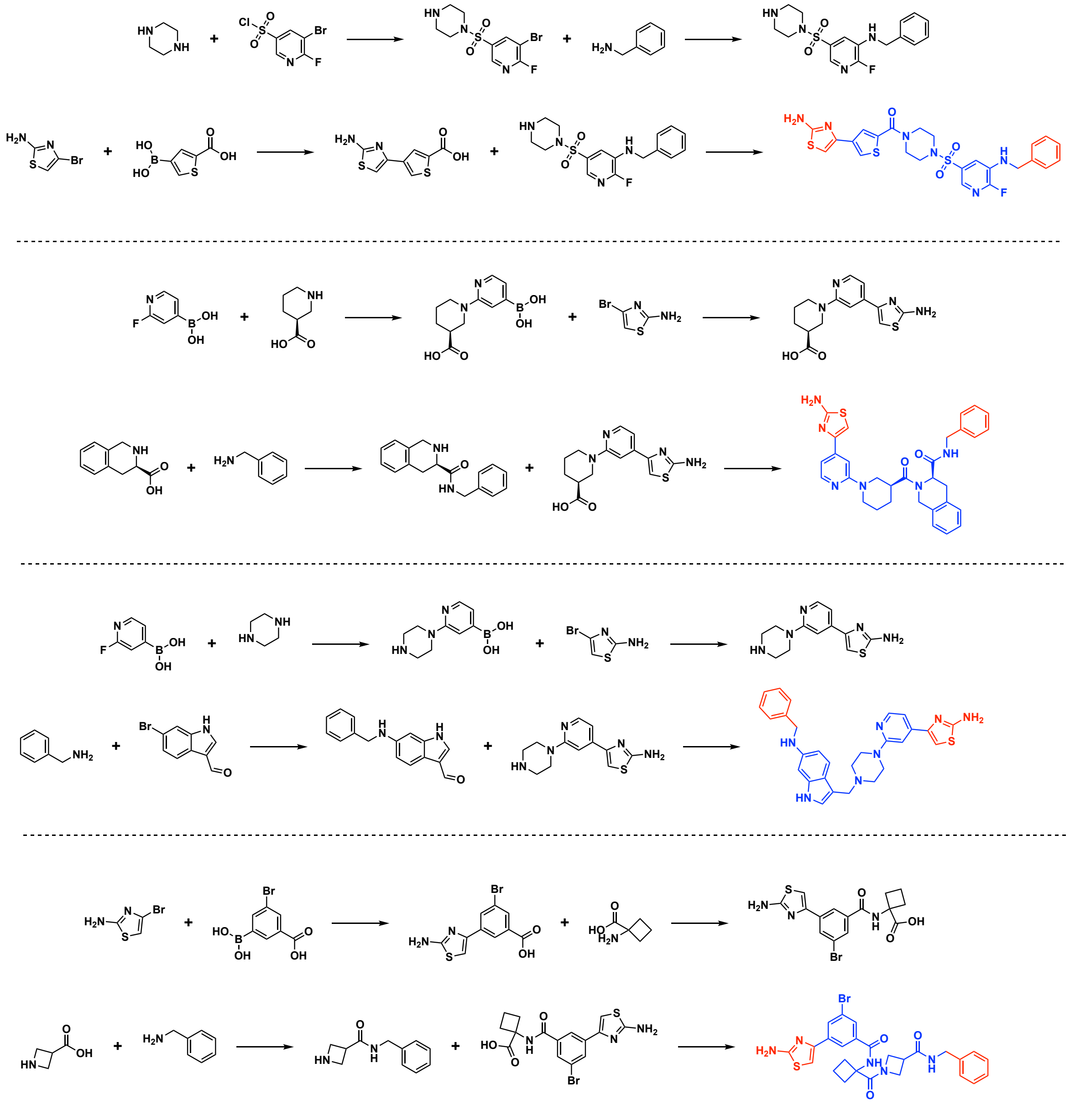

Figure 10. Synthetic pathways for molecules generated in the molecular inpainting task for target 3NDX/4EYR. The final product is shown in blue, the inpainted fragments are shown in red.