CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Generating Molecules in 3D from a Single Sequence

An Autoregressive Model with Interleaved SMILES and Internal Coordinates

Master's thesis in Computer science and engineering

Yuwei Shi & Jinyi Zhao

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2025

Master's thesis 2025

Generating Molecules in 3D from a Single Sequence

An Autoregressive Model with Interleaved SMILES and Internal Coordinates

Yuwei Shi & Jinyi Zhao

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2025

An Autoregressive Model with Interleaved SMILES and Internal Coordinates
Yuwei Shi & Jinyi Zhao

Academic Supervisor: Simon Olsson, CSE
Company Supervisors: Alessandro Tibo & Ross Irwin, Astrazeneca
Examiner: Ola Engkvist, CSE & Astrazeneca

Master's Thesis 2025
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Gothenburg, Sweden 2025

An Autoregressive Model with Interleaved SMILES and Internal Coordinates
Yuwei Shi & Jinyi Zhao
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

The de novo generation of three-dimensional molecular structures is a fundamental task in drug discovery, where state-of-the-art approaches often rely on computationally expensive and architecturally complex SE(3)-equivariant models. This thesis explores a simpler, representation-centric paradigm. We introduce a novel method that uses a standard, non-equivariant autoregressive Transformer to generate molecules from a single, unified sequence. This sequence is constructed by interleaving discrete tokens for chemical topology (from SMILES) with discretized tokens for 3D geometry (from internal coordinates), reframing the entire task as a pure language modeling problem. Our primary discrete model, ALT_TOKEN, demonstrates the success of this strategy, achieving 99.0% chemical validity and generating structures with a low median energy of 3.07 kcal/mol that closely match the dataset distribution. These results outperform baselines using continuous representations. In conclusion, this work establishes that a standard Transformer, when paired with a carefully designed discrete and interleaved data representation, provides a viable, efficient, and less complex alternative for high-quality 3D molecular design.

Keywords: Molecular generation, Internal coordinates, Language models.

Acknowledgements

We would like to express our sincere gratitude to the many people who supported us throughout our Master's thesis journey. We extend our deepest thanks to our supervisors: our academic supervisor, Professor Simon Olsson, for his invaluable guidance and scientific direction; and our company supervisors at AstraZeneca, Alessandro Tibo and Ross Irwin, for their crucial technical expertise and industry perspective. We are also very grateful to our examiner, Professor Ola Engkvist, for his careful review and insightful feedback. Our thanks go to Chalmers University of Technology, AstraZeneca, and the Molecular AI team for providing a stimulating and supportive research environment. Finally, on a personal note, we offer our heartfelt thanks to our families and friends for their love, support, and encouragement, which were essential to the completion of this work.

Yuwei Shi & Jinyi Zhao, Gothenburg, 2025-09-30

Contents

List of Figures

List of Tables

1

Introduction

The discovery of novel molecules with specific, desired properties plays a central role in advancing scientific research, particularly in fields like drug discovery [1] and materials science [2]. However, the sheer scale of the potential chemical space, estimated to contain up to 10^{60} drug-like molecules [3], makes exhaustive experimental exploration computationally infeasible, thereby motivating the development of efficient computational strategies for de novo molecular design [4].

A major challenge in this field is the accurate generation and representation of the molecular 3D structure. Although 1D string representations (such as SMILES) [5], [6] and 2D graphs are widely used in molecular language modeling [7], [8], they lack explicit 3D information, which is essential to capture molecular function, binding affinity, and physical plausibility [9]–[12]. This has motivated the development of generative models that directly produce 3D structures, utilizing frameworks such as normalizing flows, variational autoencoders, diffusion models, and autoregressive methods [13]–[16].

Recent years have seen a focus on SE(3)-equivariant architectures, where models are explicitly designed to respect rotational and translational symmetries. In practice, this means that if the input molecular coordinates are rotated or translated, the models outputs transform consistently. Such architectures, usually implemented via graph neural networks or equivariant flows, are hypothesized to promote the generation of 3D structures that are both physically meaningful and generalizable [13], [14]. Nevertheless, such models are often complex and computationally intensive, and even state-of-the-art architectures may require additional post-processing to ensure structural validity [17], [18].

Recent advances in adjacent domains provide an alternative perspective. For example, simplified models such as standard Transformers have shown strong performance in protein structure prediction and backbone generation, sometimes without relying on explicit symmetry constraints or detailed domain-specific priors [19]–[21]. These results raise the following questions: how much architectural complexity and symmetry bias are actually needed for effective 3D molecular generation? Can simple, non-equivariant models perform well if given appropriate representations?

The paradigm for 3D generative modeling is itself under active investigation. Diffusion models are widely regarded as the standard for continuous 3D structure generation [14], [22], [23]. In contrast, autoregressive models generate molecules sequen-

tially, atom-by-atom or fragment-by-fragment, which naturally allows for molecules of arbitrary size, supports exact likelihood computation, and can offer more efficient sampling [11], [24]–[26]. However, it has been suggested that autoregressive models are less effective for 3D molecular generation, due to the challenge of representing continuous spatial information as sequences.

To address this, recent work has explored hybrid autoregressive models that interleave discrete and continuous information. For example, the Quetzal model alternates between predicting the atom type (a discrete token) and its continuous 3D position, showing that this approach can bridge the gap between the flexibility of autoregressive frameworks and the geometric fidelity required for molecular structures [27], [28]. Building on this idea, we ask ourselves the following research question: can standard autoregressive models generate valid and diverse 3D molecules by alternating between atom type tokens and internal coordinates, potentially using either discrete or continuous representations?

In this work, we systematically investigate these questions by developing an autoregressive framework that alternately generates atomic types (SMILES tokens) and geometric features, represented either as discretized or continuous internal coordinates (bond lengths, angles, and dihedrals). Internal coordinates are inherently invariant to global rotation and translation, and may offer a compact and physically meaningful way to encode geometry [11], [15]. By combining these with chemical topology in an interleaved sequence, we aim to test the limits of minimalist, non-equivariant Transformer models for 3D molecular generation, and to better understand the trade-offs between representational choices, model simplicity, and generative quality.

Given practical and experimental constraints, our study focuses on the feasibility and potential limitations of this approach, including chemical validity, geometric accuracy, and computational efficiency. Although comprehensive benchmarking against state-of-the-art methods remains as a future work, our results provide an initial assessment of the strengths and open challenges for autoregressive, sequence-based 3D molecular generation with hybrid representations.

Our main contributions are:

- We introduce an autoregressive framework that alternates between generating atomic types (SMILES tokens) and internal coordinates (in either discretized or continuous form), which are naturally invariant to global rotations and translations, thus enabling 3D molecular structure generation with standard Transformer architectures and without explicit symmetry constraints.
- We analyze the impact of representation design, such as token ordering, coordinate discretization, and the interplay between topology and geometry, on model performance and generalization.
- We evaluated the strengths and limitations of this approach in terms of validity, geometric accuracy, and efficiency (definitions of which will be provided later), offering new insights into the capabilities and challenges of minimalist autoregressive models for 3D molecular generation.

2

Background

2.1 Molecular Representations

The choice of molecular representation is fundamental to the capabilities and design of any generative model. It dictates how structural and chemical information is encoded and processed by the model. In this section, we will introduce all the main molecular representations, including the specific representations used in this thesis.

2.1.1 SMILES

The Simplified Molecular Input Line Entry System (SMILES) [5] is a versatile and compact molecular representation. Representing molecules as strings of ASCII characters, SMILES supports a wide array of molecular arrangements, including branching, ring structures, and stereochemistry, through straightforward rules and symbols.

When a SMILES string is generated, it walks through the molecular graph in a depth-first search (DFS) manner [29]. Starting from an arbitrarily chosen root atom, it follows one bond at a time as far as possible, by diving down a branch until it reaches a terminal atom, before backtracking to the most recent branching point and exploring the next branch. This DFS ordering is what determines the linear sequence of atoms and parentheses in the final SMILES string. While this traversal rule is consistent, it still permits many alternative strings: the root atom can be chosen differently, branches can be listed in various orders, and ring closures can be placed at different points. Consequently, a single molecule can have numerous valid SMILES representations. Canonicalization algorithms [30] resolve this ambiguity by selecting a unique root and branch order (often guided by atom-ranking schemes), thus producing one deterministic 'canonical' SMILES for each distinct molecular graph. In this thesis, we consider only canonical SMILES to eliminate redundancy and facilitate consistent molecular encoding.

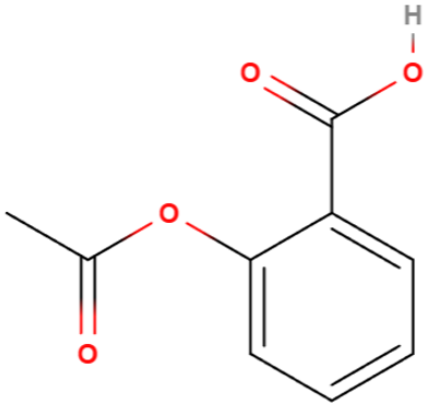


Figure 2.1: Molecule graph for aspirin (top) and its canonical SMILES (bottom). Black characters represent carbon atoms, green characters represent non-atom symbols, and red characters represent oxygen atoms.

2.1.2 Graph Representations in 3D Modeling

Graph-based representations, in which atoms are represented as nodes and bonds as edges, constitute a widely adopted alternative for 3D molecular modeling [31]. Such approaches explicitly encode atomic connectivity and readily integrate 3D coordinate information as node or edge features.

2.1.3 Cartesian Coordinates

These provide a direct and unambiguous description of each atom's position in 3D Euclidean space. However, models that process Cartesian coordinates directly must address the SE(3) symmetry group (translations and rotations in 3D space). To ensure that predictions are physically meaningful (i.e., rotating the input molecule results in a consistently rotated output distribution), such models often require architectures with built-in SE(3) equivariance [32].

2.1.4 Internal coordinates

An alternative is to use internal coordinates, typically defined sequence of tuples, where each tuple (b, θ, φ) contains of a bond length ($b \in \mathbb{R}_{\geq 0}$), a bond angle ($\theta \in (0, \pi]$), and a dihedral or torsional angle ($\varphi \in (-\pi, \pi]$) that specify molecular geometry in a relative, atom-by-atom manner. A key advantage of internal coordinates is their inherent invariance to rigid translations and rotations; the values of bond lengths, bond angles, and dihedral angles remain unchanged when the molecule undergoes any rigid-body movement in space. This property can simplify

the symmetry requirements of the generative model itself, allowing for more generic models.

While offering chemically intuitive parameters, internal coordinates present their own challenges, such as defining a canonical atom ordering for complex or cyclic molecules (often addressed by strategies like depth-first search, breadth-first search, or rule-based chemical heuristics) and ensuring valid ring closures when generating structures atomistically [33], [34].

2.1.4.1 Z-matrix

The Z-matrix is a well-established method for describing molecular systems through their internal coordinates. This approach facilitates the definition of a molecule's geometry by specifying parameters such as bond lengths, bond angles, and dihedral angles. Its construction process, where each new atoms position is defined relative to a set of previously defined atoms (e.g., an atom, an atom its bonded to, and an atom bonded to that second atom, forming a basis for angle and dihedral definitions), naturally lends itself to a sequential, autoregressive definition. This aligns well with sequence modeling paradigms like Transformers.

Let $\mathcal{A} = \{1, \dots, N\}$ be the index set of atoms and $\mathcal{E} = \{\mathrm{H}, \mathrm{C}, \mathrm{N}, \dots\}$ the set of element symbols. A Zmatrix is a map

$$\mathcal{Z} : \mathcal{A} \longrightarrow \mathcal{E} \times \mathcal{A} \times (0, \infty) \times \mathcal{A} \times (0, \pi) \times \mathcal{A} \times (-\pi, \pi],$$

$$\mathcal{Z}(a) = \Big(e_a,\ i_a,\ b_a,\ j_a,\ \theta_a,\ k_a,\ \varphi_a \Big), \qquad a \in \mathcal{A},$$

where the components have the following meaning:

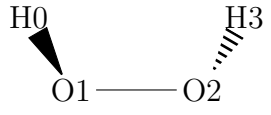
- $e_a \in \mathcal{E}$ element label of atom a;
- $i_a, j_a, k_a \in \mathcal{A}$ index of the reference atoms with $i_a \neq a$ and $j_a, k_a \neq i_a$;
- $b_a \in (0, \infty)$ *bond length*, $\theta_a \in (0, \pi)$ *bond angle*, $\varphi_a \in (-\pi, \pi]$ *torsional angle*. The calculation method will be discussed in the next section.

Using the shorthand $i \equiv i_a,\ j \equiv j_a,\ k \equiv k_a$, each row of the Zmatrix can be written compactly as

$$\Big(e_a,\ i,\ b,\ j,\ \theta,\ k,\ \varphi \Big),$$

which follows the conventional order *element, atom i, bond length b, atom j, bond angle θ, atom k, dihedral φ.*

To illustrate, consider the molecule hydrogen peroxide. A possible Z-matrix representation is:

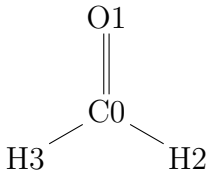



```
H0
O1 1 0.9
O2 2 1.4 1 105.0
H3 3 0.9 2 105.0 1 120.0
```

This Z-matrix begins with a hydrogen atom. The following line details an oxygen atom with a bond length of 0.9 Å to the hydrogen. Next, another oxygen is positioned 1.4 Å away from the previous oxygen with an O-O-H bond angle of 105°.

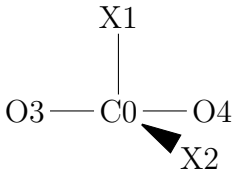
The final line describes a second hydrogen atom bonded to the second oxygen with a dihedral angle of 120°.

Similarly we have Z-matrix for a formaldehyde molecule.




```
C0
O1 1 1.2
H2 1 1.08 2 120.0
H3 1 1.08 2 120.0 3 180.0
```

As a special case, for linear molecules the Z-matrix requires an additional dummy atom to define a reference plane, as illustrated below.




```
C0
X1 1 1.0
X2 1 1.0 2 90.0
O3 1 OC 2 90.0 3 90.0
O4 1 OC 2 90.0 3 -90.0

OC 1.2
```

As this case shows[35], while performing optimizations and dealing with angles equal to 180 degrees, we can set variables (`OC 1.2`) and use dummy nodes (`X1 X2`) to construct the Z-matrix. However, in our situation, we do not perform optimization, and achieving an exact 180-degree angle is uncommon due to the conversion loss from 3D to internal coordinates. Therefore, a simple Z-matrix without dummy nodes and variables is sufficient for our purposes.

2.1.4.2 Natural extension Reference Frame (NeRF)

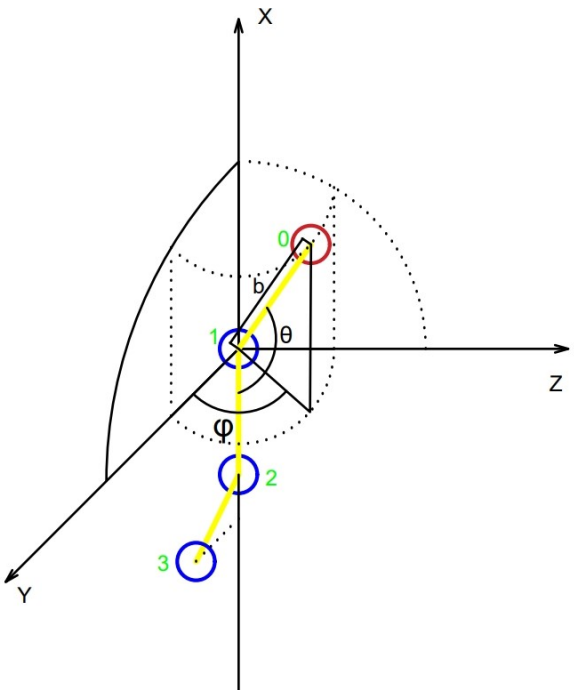


Figure 2.2: Illustration of the NeRF algorithm.

The Natural Extension Reference Frame (NeRF) algorithm [36] is a deterministic procedure to convert internal coordinates, specifically bond lengths, bond angles, and torsion angles, into Cartesian coordinates. This NeRF is unrelated to the Neural Radiance Fields widely used in computer vision, and the distinction should be made explicit to avoid confusion.

The NeRF algorithm is shown in figure 2.2 places a new atom (shown in red) using three previously placed reference atoms (shown in blue). The bond connecting atoms is represented in yellow.

We number the atom to be defined as atom 0, the closest reference atom as atom 1, followed by atoms 2 and 3. We place our origin of the local coordinate system on atom 1, make the atom 2 align with the negative x axis direction and atom 3 on the plane xy.

Let $\boldsymbol{r}_i$, $i \in \{0, 1, 2, 3\}$ be the cartesian coordinates of the atoms 0, 1, 2 and 3. Define the relative position vector as $\boldsymbol{r}_{i,j} = \boldsymbol{r}_j - \boldsymbol{r}_i$, and the normalized direction vector as $\hat{\boldsymbol{r}}_{i,j} = \frac{\boldsymbol{r}_{i,j}}{\|\boldsymbol{r}_{i,j}\|}$, where $\|\cdot\|$ is the norm. From these, we can define the unit normal vector to the plane formed by atoms 0, 1, 2 as

$$\hat{\boldsymbol{n}}_{0,1,2} = \frac{\boldsymbol{r}_{0,1} \times \boldsymbol{r}_{1,2}}{\|\boldsymbol{r}_{0,1}\|\|\boldsymbol{r}_{1,2}\|}, \tag{2.1}$$

where $\times$ is the cross product, and the unit normal to the plane 1, 2, 3 as

$$\hat{\boldsymbol{n}}_{1,2,3} = \frac{\boldsymbol{r}_{1,2} \times \boldsymbol{r}_{2,3}}{\|\boldsymbol{r}_{1,2}\|\|\boldsymbol{r}_{2,3}\|}. \tag{2.2}$$

We now define the standard internal coordinates. The bond length between atoms i and j is

$$b_{i,j} = \|\boldsymbol{r}_{i,j}\|, \tag{2.3}$$

the bond angle formed by atoms i, j, and k, where atom j serves as the pivot. With $\theta \in (0, \pi]$ as

$$\theta_{i,j,k} = \arccos\left(\frac{\boldsymbol{r}_{j,i} \cdot \boldsymbol{r}_{j,k}}{\|\boldsymbol{r}_{i,j}\|\|\boldsymbol{r}_{j,k}\|}\right), \tag{2.4}$$

and the torsional angle formed by plane ijk and jkl where the axis formed by atom j and k serves as the torsion axis as

$$\varphi_{i,j,k,l} = \arctan\left(\frac{\hat{\boldsymbol{n}}_{i,j,k} \times \hat{\boldsymbol{n}}_{j,k,l} \cdot \hat{\boldsymbol{r}}_{j,k}}{\hat{\boldsymbol{n}}_{i,j,k} \cdot \hat{\boldsymbol{n}}_{j,k,l}}\right). \tag{2.5}$$

Finally, if we place the origin at atom 1, with atom 2 on the negative x-axis and atom 3 lying in the xy-plane as we defined before, the position of atom 0 in the local reference frame is given by

$$\boldsymbol{r}_{local} = (b\cos(\theta),\ b\cos(\varphi)\sin(\theta),\ b\sin(\varphi)\sin(\theta)) \tag{2.6}$$

where $b = b_{0,1}$, the bond length is formed by atom 0 and 1. Similarly, $\theta = \theta_{0,1,2}$ means the bond angle is formed by the bond between atom 0, 1 and bond between atom 1, 2, $\varphi = \varphi_{0,1,2,3}$ represent the torsional angle between planes formed by atom 0, 1, 2 and atom 1, 2, 3.

This is the position in the local coordinate system. To obtain the global coordinates, we apply a rotation matrix $\boldsymbol{M}$:

$$\boldsymbol{M} = [\hat{\boldsymbol{r}}_{1,2},\ \hat{\boldsymbol{n}}_{0,1,2} \times \hat{\boldsymbol{r}}_{1,2},\ \hat{\boldsymbol{n}}_{0,1,2}] \tag{2.7}$$

Given the position of atom 3 in the global frame $\boldsymbol{r}_3$, the global coordinates of atom 0 are

$$\boldsymbol{r}_{global} = \boldsymbol{M}\boldsymbol{r}_{local} + \boldsymbol{r}_3, \tag{2.8}$$

This final transformation depends solely on the positions of the reference atoms 1, 2, and 3, and is independent of the internal bond parameters.

2.1.4.3 Ordering Method

The Natural extension Reference Frame (NeRF) procedure described in the previous section reconstructs the Cartesian coordinates of a new atom a from three previously placed reference atoms (r_1, r_2, r_3). To guarantee that these references exist for every atom of an N-atom molecule, we first impose a deterministic numbering $\pi = (\pi_0, \ldots, \pi_{N-1})$ on the molecular graph and then compile, from that ordering, a reference matrix $R \in \{-1, \ldots, N-1\}^{N\times 3}$. Row $R[a]$ lists the indices of up to three atoms that have already been placed when atom a is encountered; these serve as the reference atoms (r_1, r_2, r_3) for NeRF placement. The sentinel value -1 is used whenever fewer than three such atoms exist.

To ensure the current atom and reference atoms are spatially close while the lengths and angles are actual internal coordinates, our ordering strategy is intentionally designed to be straightforward: we first choose an arbitrary root atom and perform a depth-first traversal of the molecular graph to obtain the sequence $\pi = (\pi_0, \pi_1, \ldots, \pi_{N-1})$. For every subsequent atom $a = \pi_k$ with $k \geq 3$, we then greedily assign its three references by expanding outward in the graph: the first reference r_1 is the nearest neighbour of a ($d(a, r_1) = 1$); the second reference r_2 is the nearest already-placed neighbour of r_1; and the third reference r_3 is the nearest already-placed neighbour of r_2. If no suitable neighbour exists at any stage, we simply fall back to the lexicographically earliest atom that has already been placed.

The complete procedure is given in Algorithm 1. Its output is immediately usable by NeRF: when atom a is processed, the triple $(r_1, r_2, r_3) = \boldsymbol{R}[a]$ supplies the *exact* atoms required to evaluate the bond length $b_{0,1}$, bond angle $\theta_{0,1,2}$, and torsion $\varphi_{0,1,2,3}$ defined in 2.7.

Algorithm 1 Build Reference Matrix

Require: Molecule $\mathcal{M}$ with N atoms; atom ordering $\pi = (\pi_0, \ldots, \pi_{N-1})$
Ensure: Reference matrix $\boldsymbol{R}[N][3]$ such that each atom has up to 3 references from previously placed atoms

$\boldsymbol{R} \leftarrow (-1, -1, -1)^N$ ▷ Initialize all entries
if $N > 1$ **then**
  $\boldsymbol{R}[\pi_1] \leftarrow (\pi_0, -1, -1)$ ▷ Set reference for second atom
end if
if $N > 2$ **then**
  $\boldsymbol{R}[\pi_2] \leftarrow (\pi_1, \pi_0, -1)$ ▷ Set references for third atom
end if
for k from 3 to $N-1$ **do**
  Let $a \leftarrow \pi_k$ ▷ Current atom
  Let $\mathcal{C} \leftarrow \{\pi_0, \ldots, \pi_{k-1}\}$ sorted by topological distance $d(a, \cdot)$
  $r_1 \leftarrow$ first $c \in \mathcal{C}$ with $d(a, c) = 1$ **else** first of $\mathcal{C}$
  $r_2 \leftarrow$ first $c \in \mathcal{C} \setminus \{r_1\}$ with $d(r_1, c) = 1$ **else** first of $\mathcal{C} \setminus \{r_1\}$
  $r_3 \leftarrow$ first $c \in \mathcal{C} \setminus \{r_1, r_2\}$ with $d(r_2, c) = 1$ **else** first of $\mathcal{C} \setminus \{r_1, r_2\}$
  Assign references: $\boldsymbol{R}[a] \leftarrow (r_1, r_2, r_3)$
end for
return $\boldsymbol{R}$

2.1.5 Assessing and Refining 3D Geometric Plausibility: Merck Molecular Force Field (MMFF94)

To quantitatively evaluate whether the generated molecular conformations are geometrically reasonable, we pick MMFF94 [37] for our molecule energy calculation. MMFF94 is a general force field that uses a fixed set of parameters to model organic molecules. Its total energy is the sum of seven independent terms:

$$\begin{aligned} E_{\text{MMFF94}} = &\sum_{\langle ij \rangle} E_{\text{bond}} + \sum_{\langle ijk \rangle} E_{\text{angle}} + \sum_{\langle ijk \rangle} E_{\text{str–bend}} \\ &+ \sum_{\langle ijkl \rangle} E_{\text{oop}} + \sum_{\langle ijkl \rangle} E_{\text{torsion}} + \sum_{i<j} E_{\text{vdW}} + \sum_{i<j} E_{\text{elec}}, \end{aligned} \tag{2.9}$$

Valence terms

- **Bond stretching** quartic expansion of a Morse potential, $E_{\text{bond}} = k_r\,(\Delta r)^2\Big[1 + c_s\,\Delta r + \frac{7}{12}c_s^2(\Delta r)^2\Big]$ with $c_s = -2\,\text{Å}^{-1}$, and k_r is the force constant.
- **Angle bending** cubic expansion, $E_{\text{angle}} = k_\theta\,(\Delta\theta)^2\Big[1 + c_b\,\Delta\theta\Big]$ with $c_b = -0.4\,\text{rad}^{-1}$, and k_θ is the force constant. a special linear form is applied for near-colinear angles.
- **Stretchbend** linear coupling of the two adjacent bonds to the enclosed angle.
- **Out-of-plane bend** Wilson angle penalty $E_{\text{oop}} = k_\chi\,\chi^2$ for trigonal centres. With k_χ as the force constant.
- **Torsion** three-term Fourier series $E_{\text{torsion}} = \frac{1}{2}\Big[V_1(1 - \cos\phi) + V_2(1 - \cos 2\phi) + V_3(1 + \cos 3\phi)\Big]$.

Non-bonded terms

- **van der Waals** buffered-14-7 potential $E_{\mathrm{vdW}} = \varepsilon_{ij}\Big[\Big(\frac{R_{ij}^0}{R_{ij}+0.07}\Big)^{14} - 2\Big(\frac{R_{ij}^0}{R_{ij}+0.07}\Big)^{7}\Big]$, with combination rules based on atomic polarisabilities.
- **Electrostatics** buffered Coulomb law $E_{\mathrm{elec}} = 332.0716\,\frac{q_i q_j}{(R_{ij}+0.05)}$ (kcal mol^{-1}, Å, unit dielectric). Partial charges are derived from transferable bond-charge increments.

Performance Fitting to $\sim$1 450 high-level *ab initio* and experimental structures gives root-mean-square errors of 0.014 Å for bond lengths, 1.2° for bond angles, 61 cm^{-1} for vibrational frequencies, and 0.4 kcal mol^{-1} for conformer energies. These results indicate that MMFF94 achieves accuracy comparable to MM3, while applicable to a broader range of chemical structures. Consequently, it may be suitable for applications such as geometry optimization, molecular dynamics, and docking studies.

2.2 Symmetry in 3D Molecular Systems: SE(3) Invariance and Equivariance

A key requirement is that the probability distributions or outputs produced by the model must be invariant to rigid-body transformations (translations and rotations) and to permutation of identical atoms. More generally, if the input is transformed by an element of the symmetry group, the output should transform accordingly; this is known as equivariance [38]. Formally, let $\mathcal{G}$ denote a symmetry group, such as SE(3) for 3D rotations and translations. A function $f : \mathcal{X} \to \mathcal{Y}$ is said to be invariant to $\mathcal{G}$ if, for all $g \in \mathcal{G}$,

$$f(g \cdot x) = f(x), \qquad \forall x \in \mathcal{X}, \tag{2.10}$$

where $g \cdot x$ denotes the group action of g on x. It is equivariant if there exists a group action on $\mathcal{Y}$ (also denoted $g\cdot$) such that

$$f(g \cdot x) = g \cdot f(x), \qquad \forall g \in \mathcal{G}, , x \in \mathcal{X}. \tag{2.11}$$

In molecular tasks, scalar quantities (e.g., energies or log-densities) should be invariant, whereas vector/tensor quantities (e.g., forces, dipoles) should be equivariant under the same group action.

If $z \sim \rho_0$ is sampled from a *G-invariant* base distribution (e.g., an isotropic Gaussian), and f is an *H-equivariant* diffeomorphism with $H \subseteq G$, then the pushforward distribution of $x = f(z)$ is *H-invariant*. Consequently, sampling from a symmetric base distribution and applying an equivariant transformation yields an invariant output distribution [39].

For 3D coordinate-based molecular representations (point clouds/graphs with coordinates), the relevant group is often SE(3) $\times$ S_N, combining rigid-body transformations and permutations among identical atoms. In contrast, sequence-only

representations such as SMILES do not exhibit SE(3) or S_N symmetries; their symmetry considerations arise only after parsing into the molecular graph (where one typically considers invariance to graph isomorphisms). Internal coordinates are intrinsically SE(3)-invariant by construction, but permutation invariance generally requires canonicalization or permutation-aware modeling, and care is needed for chirality and signed torsions.

In the following, we review two main strategies for achieving SE(3) consistency: building equivariant neural network architectures, and employing invariant representations at the data level.

2.2.1 Equivariant Neural Network Architectures

One approach for ensuring SE(3) symmetry is to build *equivariant* neural network architectures, explicitly embedding symmetry constraints into network layers. Such architectures guarantee that if the input molecular coordinates are rotated or translated, the predictions from the model transform accordingly. Prominent examples include E(n)-equivariant message-passing networks such as EGNNs [40], and irreducible-representation-based models such as Tensor Field Networks (TFNs) [41] and SE(3)-Transformers [42], which explicitly encode SE(3) transformations into their computational structures. While this approach rigorously enforces symmetry, it often introduces additional complexity into network design. Operations must be carefully structured around equivariant tensor computations, potentially leading to higher computational costs, increased model complexity, and reduced flexibility in architecture design.

2.2.2 Invariant Input Representations

An alternative and simpler strategy is to use invariant input representations, such as internal coordinates (bond lengths, bond angles, dihedral angles). Internal coordinates naturally remain unchanged under rigid-body translations and rotations [43]. *Permutation invariance is not automatic*: it requires a canonicalization procedure (e.g., fixed traversals and reference-frame conventions) or a permutation-aware model; additional care is needed for chirality and signed torsions. As a result, standard neural architectures, including Transformer-based sequence models, can often process such representations without specialized equivariant layers. By encoding invariance at the representation level, this approach enables the reuse of general-purpose sequence modeling techniques originally developed in domains like natural language processing. These models benefit from mature infrastructure, well-understood training procedures, and existing optimizations for Transformers [23]. However, this simplification introduces new challenges. One major difficulty lies in the construction and canonical ordering of internal coordinates, especially for cyclic or highly branched molecules. Researchers typically use graph traversals (e.g., depth-first search) or domain-specific heuristics to define atom orderings and reference frames [43]. Although this makes internal coordinate-based models viable in many cases, scalability and robustness across diverse molecular classes remain active areas of investigation.

2.2.3 Choosing Between Equivariance and Invariance

The choice between equivariant architectures and invariant input representations ultimately depends on the applications specific requirements and constraints. Equivariant architectures offer rigorous symmetry preservation at the cost of architectural complexity and computational overhead. In contrast, invariant representations simplify architectures and training, leveraging standard modeling tools, with challenges shifted toward careful data preprocessing and representation design. Given these considerations, this thesis primarily adopts an invariant representation approach, focusing on internal coordinates to embed symmetry constraints directly within the data, thus enabling the effective and efficient use of Transformer-based generative models.

2.3 The Transformer as a Generative Model

2.3.1 Transformer Architecture

Traditional sequence models, such as recurrent neural networks (RNNs), process input sequences sequentially, using hidden states to capture dependencies between tokens. In contrast, the Transformer architecture [44] is based on self-attention mechanisms, allowing each token to directly attend to all other tokens in a sequence. This design enables more efficient parallelization and facilitates the modeling of long-range dependencies. The original Transformer model consists of encoder and decoder components, but for autoregressive sequence generation, typically only the decoder is used, generating one token at a time conditioned on previously generated tokens.

2.3.1.1 Self-Attention Mechanism

The core of the Transformer is the self-attention mechanism. Given an input sequence of n token embeddings $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \in \mathbb{R}^{n \times d}$, self-attention computes a new set of representations $\mathbf{Z} \in \mathbb{R}^{n \times d}$, where each output $\mathbf{z}_i$ is a weighted sum of all input embeddings:

$$\mathbf{z}_i = \sum_{j=1}^{n} \alpha_{ij} (\mathbf{x}_j \mathbf{W}^V) \tag{2.12}$$

where the attention weights α_{ij} are computed as:

$$\alpha_{ij} = \frac{\exp\left(\frac{(\mathbf{x}_i \mathbf{W}^Q)(\mathbf{x}_j \mathbf{W}^K)^\top}{\sqrt{d_k}}\right)}{\sum_{k=1}^{n} \exp\left(\frac{(\mathbf{x}_i \mathbf{W}^Q)(\mathbf{x}_k \mathbf{W}^K)^\top}{\sqrt{d_k}}\right)} \tag{2.13}$$

Here, $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V \in \mathbb{R}^{d \times d_k}$ are learnable parameter matrices for queries, keys, and values, respectively. The scaling factor $\sqrt{d_k}$ stabilizes gradients during training.

In the decoder, a causal (or masked) self-attention is used to prevent the prediction of a token from attending to future positions.

2.3.1.2 Multi-Head Attention

To allow the model to jointly attend to information from different representation subspaces, the Transformer employs multi-head attention. Specifically, the input embeddings are projected into h distinct subspaces (heads), and attention is performed in each:

$$\text{MultiHead}(\mathbf{X}) = [\text{head}_1; \ldots; \text{head}_h]\mathbf{W}^O \tag{2.14}$$

$$\text{head}_i = \text{Attention}(\mathbf{X}\mathbf{W}_i^Q, \mathbf{X}\mathbf{W}_i^K, \mathbf{X}\mathbf{W}_i^V) \tag{2.15}$$

where $[\cdot;\cdot]$ denotes concatenation and $\mathbf{W}^O$ is an output projection matrix. This mechanism enables the model to capture diverse relationships and dependencies within the sequence.

2.3.1.3 Feedforward Network, Residual Connections, and Layer Normalization

Each sub-layer in the Transformer is followed by a residual connection [45] and layer normalization [46] to facilitate gradient flow and stable training. After the attention mechanism, each token embedding passes through a position-wise feedforward network:

$$\text{FFN}(\mathbf{x}) = \max(0,\, \mathbf{x}\mathbf{W}_1 + \mathbf{b}_1)\,\mathbf{W}_2 + \mathbf{b}_2 \tag{2.16}$$

where $\mathbf{W}_1, \mathbf{W}_2$ are learnable weights and $\mathbf{b}_1, \mathbf{b}_2$ are biases. This component increases model expressiveness and enables nonlinear transformations.

2.3.1.4 Positional Encoding

Since self-attention is permutation-invariant, positional information must be injected to allow the model to distinguish token order. In the original Transformer, this is achieved using sinusoidal positional encodings:

$$\text{PE}_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d}}\right) \tag{2.17}$$

$$\text{PE}_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d}}\right) \tag{2.18}$$

where pos is the position and i indexes the dimension. Alternatively, positional embeddings can be learned as additional parameters.

2.3.2 The Autoregressive Generation Process

2.3.2.1 Introduction

Having detailed the static components of the Transformer architecture in the previous chapter, including the self-attention mechanism and positional encoding, we now shift focus from the structure of the model to its application in generative

tasks. Among its various uses, the Transformer has been widely adopted as an auto-regressive model for sequence generation tasks. This chapter will introduce the fundamental principles of auto-regressive sequence generation and describe the mechanisms, such as masked self-attention, that enable the Transformer to produce complex sequences token by token. These details provide the necessary background for our specific approach to molecular generation.

2.3.2.2 The Principle of Autoregressive Modeling

An autoregressive model generates a sequence of data, $\mathbf{x} = (x_1, ..., x_T)$, by modeling the conditional probability of each element given the elements that preceded it. This process follows the chain rule of probability, allowing the joint probability distribution of the entire sequence to be factorized into a product of conditional probabilities:

$$p(\mathbf{x}) = p(x_1, ..., x_T) = \prod_{t=1}^{T} p(x_t|x_{<t}) \tag{2.19}$$

where x_t is the token at timestep t, and $x_{<t}$ represents the sequence of all preceding tokens $(x_1, ..., x_{t-1})$.

This principle is perhaps most intuitively understood through the lens of natural language processing. When a language model generates a sentence, it predicts the next word based on the words it has already written. For instance, given the context "The cat sat on the," the model calculates a probability distribution over its vocabulary to predict the most likely next word, such as "mat." This step-by-step, conditional generation is the essence of autoregression.

2.3.2.3 Masked Self-Attention: Enabling Autoregression in Transformers

To enforce this autoregressive property within the Transformer architecture [44], a crucial modification is made to the self-attention mechanism. This modification is known as masked self-attention.

The mechanism works by applying a look-ahead mask or causal mask to the attention scores before the softmax operation is applied. This mask effectively sets the attention scores for all subsequent positions to negative infinity ($-\infty$). After the softmax function is applied, these scores become zero, ensuring that any given position t can only attend to previous positions (from 1 to $t-1$) and itself. This prevents any information from "leaking" from future tokens into the prediction of the current token, thereby strictly preserving the conditional dependency $p(x_t|x_{<t})$ that defines the autoregressive model.

2.3.2.4 The Full Generation Process (Inference)

At inference time, the trained autoregressive Transformer generates a new sequence token by token in a sequential loop. The process begins with a special start-of-sequence token, denoted ‘[BOS]‘. This initial token is fed into the model, which then

produces a probability distribution, in the form of logits, over its entire vocabulary for the next token in the sequence.

From this distribution, a token is selected using a chosen decoding strategy. Common strategies include greedy decoding, which always selects the most probable token; beam search, which maintains a set of k most probable sequences at each step; or stochastic sampling methods such as top-k or nucleus sampling, which samples from a truncated probability distribution to increase diversity.

Once a token is selected, it is appended to the current sequence. This new, longer sequence then becomes the input for the next time step. This loop of 'predict -> sample -> append' continues until the model generates a special end-of-sequence token, '[EOS]', or a predefined maximum sequence length is reached.

2.3.2.5 Application to Our Work: From Text to Molecules

Now that we have established the general framework for how a Transformer generates a sequence, we can apply this to our specific task. In our work, the sequence to be generated is not composed of natural language words but rather our novel interleaved representation of a molecule. Consequently, the model's vocabulary is a composite set containing both discrete tokens for chemical topology (from SMILES) and discrete tokens or continuous coordinates representing 3D geometry.

Therefore, the central task of our model is to learn the complex conditional probability distribution of this specialized "molecular language". It learns to predict the next token in the sequence, be it an atom or a group of bond length, bond angle, and dihedral, based on the rich topological and geometric context provided by all preceding tokens. This sequential formulation enables us to treat 3D molecular generation as a sequence modeling problem and apply standard autoregressive modeling techniques to this task.

2.3.3 Related Work

Generative modeling of 3D molecular structure has progressed along several complementary lines. Early successes were achieved with diffusion-based approaches, which refine Gaussian noise atomic coordinates through a learned reverse process [14]. Variants that embed SE(3)-equivariant graph networks have set strong baselines for geometric fidelity, and scaling trends observed in large-parameter models such as Megalodon suggest that capacity can further improve topological stability [47]. At the same time, latent diffusion formulations compress molecular structures into lower-dimensional representations before denoising, offering a pragmatic route to lower computational cost [15]. However, iterative sampling remains expensive and chemical validity is not yet guaranteed without additional post-processing [48]. These practical limitations have motivated interest in alternatives that reduce inference time while maintaining geometric accuracy.

One such alternative is flow-matching, where an ordinary differential equation is learned to transport a simple prior directly to the data distribution. Models such as FlowMol and SemlaFlow demonstrate that deterministic flows can shorten inference

by an order of magnitude relative to diffusion, yet reach comparable metrics on standard benchmarks [16], [49]. Nevertheless, the requirement for symmetry-aware backbones and the difficulty of satisfying hard chemical constraints indicate that further simplification might be beneficial.

A separate thread explores autoregressive generation, building molecules atom-by-atom or fragment-by-fragment. Early autoregressive models discretised Cartesian coordinates and struggled to capture spatial continuity, but more recent hybrids alternate discrete atom types with continuous coordinate updates to good effect. Quetzal, for instance, couples a Transformer that predicts the next atom identity with a lightweight diffusion module that places the atom in space, narrowing the quality gap to diffusion while retaining fast sampling and exact likelihoods [27]. Two-stage pipelines such as NExT-Mol generate a 2-D graph with a language model before predicting 3-D conformations with an equivariant network, illustrating that topology and geometry can be learned by specialized components [50]. These studies collectively suggest that sequential decoding remains a viable option provided that chemical and spatial information are kept in step during generation.

Attention has also turned to the role of the Transformer itself. Equivariant variants embed rotational symmetry directly in the architecture, whereas non-equivariant Transformers rely on data representation or data augmentation to learn invariances implicitly [51]. Recent work indicates that with suitable relative positional encodings or learned canonical frames, standard Transformers can approach the performance of more specialized designs [52]. This observation motivates testing whether an invariant input description alone can relieve the need for symmetry-aware layers, especially when one wishes to reuse efficient implementations developed for natural language processing.

The effectiveness of any architecture would also depends on how the 3D information is encoded. Cartesian coordinates are straightforward, but require the network to learn Euclidean symmetries; voxel grids support convolutional inductive biases but incur high memory costs and an additional reconstruction step; internal coordinates remove global rotations and translations by construction, but introduce challenges of ordering and error propagation along kinematic chains [53]. Discretising internal coordinates into token vocabularies aligns naturally with language models but sacrifices numerical precision, whereas continuous values preserve detail at the expense of a larger regression burden [26]. Vector-quantised codebooks offer a middle ground by learning data-driven geometric alphabets, although they add another layer of complexity to the pipeline.

The literature shows a spectrum ranging from symmetry-rich yet complex architectures to symmetry-agnostic but data-efficient designs. It remains unclear whether high-quality 3D molecules can be generated by a standard Transformer that alternates chemical topology with invariant internal coordinates, and whether discretised or continuous geometry serves this goal best. The present work evaluates this possibility, comparing discretised and continuous internal-coordinate schemes and examining how early fusion of topology and geometry influences validity, accuracy, and efficiency.

3

Methods

3.1 Dataset

The **GEOM** dataset [10] provides high-quality molecular conformations and corresponding energies. It contains ∼ 37 million 3D geometries for more than 450,000 unique molecules, generated by extensive conformer sampling followed by semi-empirical and density-functional refinements. GEOM is divided into two main parts: (i) *GEOM-QM9*, comprising ∼133,000 small organic species originally from the QM9 benchmark, and (ii) *GEOM-Drugs*, a set of ∼317,000 drug-like molecules curated from ChEMBL[54], ZINC[55], and related experimental sources. Each conformer is sampled based on the semi-empirical extended tight-binding method (GFN2-xTB [56]) to generate reliable and accurate structures. Thanks to its scale, chemical diversity, and rigorous geometry optimization using GFN2-xTB, GEOM-Drugs has become a foundational benchmark for training and evaluating 3D molecular generative models [57]. Accordingly, we adopt GEOM-Drugs as the dataset in this thesis.

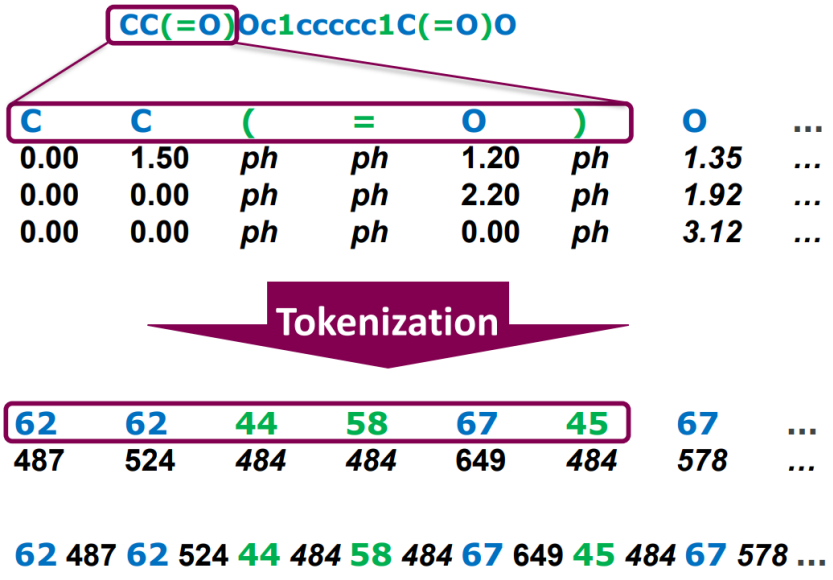


Figure 3.1: Tokenization process overview(ph for place holder)

3.2 Data Preprocessing Strategies

Molecular data are obtained from SDF files, which provide the full 3D atomic coordinates for each molecule. For every molecule, we tokenize the SMILES string with a predefined vocabulary; and compute its internal coordinates, which are bond length, bond angle, and torsion angle for each atom using our ordering method discussed in the previous section. For continuous coordinate values, we first represent all the angles with we use then discretise them into string keys (e.g. `0.95_1.57_3.14`).

These keys are subsequently mapped to token IDs through a frequency-pruned coordinate vocabulary. To preserve the alternating *token/coordinate* pattern expected by the model, every non-atom SMILES token is immediately followed by a fixed placeholder.

3.2.1 Molecule Loading and Cleaning

The workflow starts from large-scale SDF collections of GEOM-Drugs datasets, where each entry encodes a three-dimensional molecular structure. Molecules are loaded using the RDKit toolkit, with hydrogen atoms explicitly removed, and chemical sanitization enabled to ensure valence and aromaticity are correct. Molecules that fail sanitization or whose canonical SMILES cannot be generated are filtered out and logged for data integrity tracking.

For each successfully loaded molecule, the canonical SMILES string is generated and used to define a unique and consistent atom ordering for downstream processing. This step ensures that all coordinate transformations and sequence operations are reproducible and consistent across the entire dataset.

3.2.2 SMILES Tokenisation

Our SMILES vocabulary is derived from REINVENT 4 [58]. To cope with any unforeseen symbol, we add an `<UNK>` token, ensuring that every character in a raw SMILES string can be represented.

We split the full vocabulary into five intuitive subsets, each reflecting a different syntactic role in the SMILES grammar:

- $V_{\text{atom}} = \{\text{H}, \text{C}, \text{O}, \text{N}, \dots\}$ atomic symbols,
- $V_{\text{stru}} = \{(,), =, 1, 2, \dots\}$ structural symbols,
- $V_{\text{spec}} = \{\texttt{<BOS>}, \texttt{<EOS>}, \texttt{<PAD>}\}$ special tokens,
- $V_{\text{mol}} = V_{\text{atom}} \cup V_{\text{stru}}$ SMILES tokens excluding specials,
- $V_{\text{smi}} = V_{\text{mol}} \cup V_{\text{spec}}$ full, padded SMILES vocabulary.

Here `<BOS>` (begin-of-sequence) and `<EOS>` (end-of-sequence) mark sentence boundaries, while `<PAD>` is a dummy filler that keeps all sequences the same length inside a batch.

A molecules canonical SMILES string is modelled as a sequence:

$$\boldsymbol{m} = (m_1, m_2, \ldots, m_l), \qquad m_i \in V_{\text{mol}},$$

so that $\boldsymbol{m} \in V_{\text{mol}}^l$. The notation S^n denotes *all* strings of length n built from an alphabet S. For aspirin, the unpadded string reads (`C`, `C`, `(`, `=`, ..., `O`).

To feed sequences into a Transformer we require a uniform length L. The padding operator

$$\mathcal{P}_{\text{smi}} \colon V_{\text{mol}}^l \longrightarrow V_{\text{smi}}^L$$

acts as follows:

1. prepend `<BOS>` and append `<EOS>` to bracket the string;
2. right-pad with `<PAD>` until the sequence reaches exactly L tokens.

Hence

$$\boldsymbol{s} = \mathcal{P}_{\text{smi}}(\boldsymbol{m}) = (\texttt{<BOS>}, m_1, \ldots, m_l, \texttt{<EOS>}, \texttt{<PAD>}, \ldots, \texttt{<PAD>}).$$

In practice L is chosen as a global bucket large enough to hold the longest molecule in our dataset. For aspirin, the padded representation becomes
(`<BOS>`, `C`, `C`, `(`, `=`, ..., `O`, `<EOS>`, `<PAD>`, ...).

Now we comes to the final stage of tokenization – map them into numbers. We therefore define the index set

$$T_{\text{smi}} = \{0, 1, \ldots, |V_{\text{smi}}| - 1\},$$

and its sequence set

$$T_{\text{smi}}^L = \Big\{(t_1, \ldots, t_L) \mid t_i \in T_{\text{smi}}\Big\}.$$

The tokenize function is a deterministic, one-to-one mapping

$$\mathcal{T}_{\text{smi}} \colon V_{\text{smi}}^L \longrightarrow T_{\text{smi}}^L$$

assigns a unique integer to each vocabulary symbol, yielding

$$\boldsymbol{t}_{\text{smi}} = \mathcal{T}_{\text{smi}}(\boldsymbol{s}) = (t_1, t_2, \ldots, t_L).$$

Continuing the example, aspirin is finally encoded as, e.g. $(0, 62, 62, 44, \ldots, 67, 1, 2, \ldots, 2)$. Here 0 represents `<BOS>`, 1 represents `<EOS>`, 2 is `<PAD>`, and indices $44, 62, 67, \ldots$ correspond to the concrete chemical and structural symbols appearing in the original string.

3.2.3 Coordinate Extraction and Tokenisation

For every molecule we first build a strict, ordering-aware Z-matrix through the NeRF algorithm discussed in Section 2.1.4. NeRF provides a deterministic, one-to-one mapping from Cartesian coordinates to internal coordinates, meaning that the same molecule will *always* yield the same Z-matrix irrespective of any global rotations or translations.

For an atom with index a the Z-matrix row is

$$\mathcal{Z}(a) = \Big(e,\, i,\, b,\, j,\, \theta,\, k,\, \varphi\Big), \qquad a \in \mathcal{A},$$

where

- e is the element type (e.g. C, N, O);
- $i, j, k \in \mathcal{A}$ are reference indices for bond, angle, and dihedral anchors;
- b (bond length), θ (bond angle), and φ (torsion angle) are the local internal coordinates with ranges $b \in (0, \infty)$ Å, $\theta \in (0, \pi)$ rad, $\varphi \in (-\pi, \pi]$ rad.

The symbol set $\mathcal{A}$ denotes all atom indices in the molecule.

To make angles numerically smooth and $-\pi \equiv +\pi$ invariant, we embed each triplet as a five-dimensional vector that mixes raw and trigonometric terms:

$$\boldsymbol{c} = \Big(d,\ \sin\theta,\ \cos\theta,\ \sin\varphi,\ \cos\varphi\Big).$$

Using sine and cosine ensures that geometrically identical angles map to the *same* point in feature space, eliminating discontinuities around $\pm\pi$.

We arrange these vectors so that they align token-by-token with the SMILES string: every atomic symbol receives its own coordinate vector, whereas every structural or padding symbol receives a zero vector of the same length. This gives a continuous sequence

$$\boldsymbol{C} = (\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_l), \qquad \boldsymbol{c}_i \in S_{\text{ics}},$$

with $S_{\text{ics}} = (0, \infty) \times [0, 1] \times [-1, 1] \times [-1, 1] \times [-1, 1]$; in shorthand, $\boldsymbol{C} \in S_{\text{ics}}^{l}$.

To feed coordinates into the same embedding table as tokens we quantise each triplet with uniform step sizes

$$\Delta d = 0.05 \text{ Å}, \quad \Delta\theta = 0.05 \text{ rad}, \quad \Delta\varphi = 0.02 \text{ rad}.$$

Each quantised triplet is encoded as a string key (e.g. `1.35|2.05|0.62`). Key frequencies are collected over the entire training set; the rarest 1% are replaced by a catch-all `<UNK>` token, leaving a compact vocabulary of 23 045 unique entries. We define

- V_{ics} quantised coordinate keys include `<UNK>` for discarded keys„
- $V_{\text{spec}} = \{\texttt{ph}\}$ placeholder,
- $V_{\text{coord}} = V_{\text{ics}} \cup V_{\text{spec}}$ full coordinate vocabulary.

The discretiser

$$\mathcal{D}\colon S_{\text{ics}}^{l} \longrightarrow V_{\text{ics}}^{l}$$

applies element-wise rounding,

$$\boldsymbol{C_d} = \mathcal{D}(\boldsymbol{C}) = (\boldsymbol{c}_{d1}, \boldsymbol{c}_{d2}, \ldots, \boldsymbol{c}_{dl}), \quad \boldsymbol{c}_{di} \in V_{\text{ics}}.$$

Because SMILES special symbols (<BOS>, <EOS>, <PAD>) carry no physical geometry, we insert a distinct placeholder ph after each such symbol to preserve the strict SMILES-coordinate alternation. The padding operator

$$\mathcal{P}_{\text{coord}} \colon V_{\text{ics}}^{\,l} \longrightarrow V_{\text{coord}}^{\,L}$$

returns

$$\boldsymbol{C_s} = \mathcal{P}_{\text{coord}}(\boldsymbol{C_d}) = (\texttt{ph}, \boldsymbol{c}_{d1}, \boldsymbol{c}_{d2}, \texttt{ph}, \ldots, \boldsymbol{c}_{dl}, \texttt{ph}, \ldots, \texttt{ph}).$$

Note that $\boldsymbol{c}_{di} = \texttt{ph}$ whenever the matching SMILES token is a structural symbol or padding token. During training we simply mask out the loss on these placeholders so that the model is never penalised for their arbitrary value.

Coordinate tokens live in a range disjoint from SMILES tokens:

$$T_{\text{coord}} = \{|V_{\text{smi}}|,\ |V_{\text{smi}}| + 1,\ \ldots,\ |V_{\text{smi}}| + |V_{\text{coord}}| - 1\}.$$

Their structured counterparts are

$$V_{\text{coord}}^{\,L} = \{(c_{s1}, \ldots, c_{sL}) \mid c_{si} \in V_{\text{coord}}\}, \quad T_{\text{coord}}^{\,L} = \{(t_{c1}, \ldots, t_{cL}) \mid t_{ci} \in T_{\text{coord}}\}.$$

Finally, the coordinate-tokeniser

$$\mathcal{T}_{\text{coord}} \colon V_{\text{coord}}^{\,L} \longrightarrow T_{\text{coord}}^{\,L}$$

produces the indexed sequence

$$\boldsymbol{t}_{\text{coord}} = \mathcal{T}_{\text{coord}}(\boldsymbol{C_s}) = (t_{c1}, t_{c2}, \ldots, t_{cL}),$$

ready to be concatenated (or alternated) with its SMILES counterpart for downstream Transformer input.

3.3 The Models

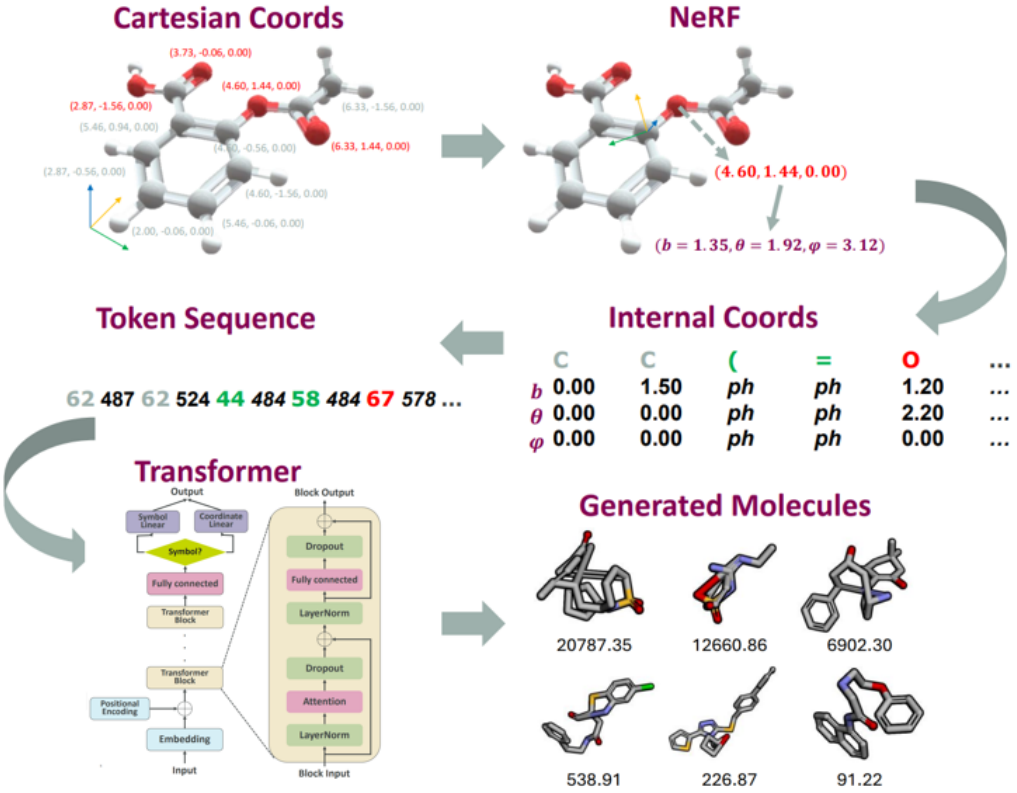


Figure 3.2: Pipeline overview. Cartesian coordinates are mapped to internal coordinates (b, θ, φ) with NeRF (§2.1.4.2). Each internal triple is discretised and interleaved with its corresponding SMILES token, forming a single sequence that is passed to the transformer decoder detailed below.

This pipeline overview highlights two guiding ideas: encoding molecular geometry in the internal coordinate system, and fusing chemistry (SMILES) and geometry (internal coordinates) into a single autoregressive stream. There are four different embedding strategies described next.

3.3.1 Embedding Strategies

We consider four input embedding schemes distinguished by (i) layoutinterleaved (ALT) vs. parallel (DE), and (ii) coordinate representationtokenised vs. continuous. The resulting variants are ALT_TOKEN, ALT_CONT, DE_TOKEN, and DE_CONT. We set the base embedding size $d_e = 128$ for SMILES and coordinate streams. In ALT we append a stream-type embedding of size $d_t = 16$ to indicate whether a position is a token or a coordinate, see details in 3.5.

Notation

- $E_{\text{smi}} : T_{\text{smi}} \to \mathbb{R}^{d_e}$ SMILES token embedding
- $E_{\text{coord}} : T_{\text{coord}} \to \mathbb{R}^{d_e}$ *tokenised* coordinate embedding
- $E_{\text{cont}} : S_{\text{ics}} \to \mathbb{R}^{d_e}$ *continuous* coordinate projection (one-layer MLP)
- $\|$ concatenation along the feature dimension

Note that the embedding function should map to $\mathbb{R}^{d_e+d_t}$ in the case of ALT model.

ALT_TOKEN: Interleaved & Tokenised

From the paired token streams $\boldsymbol{t}_{\text{smi}}$ and $\boldsymbol{t}_{\text{coord}}$ we build a single interleaved sequence

$$\boldsymbol{t}_{\text{inT}} = (t_1, t_{c1}, t_2, t_{c2}, \ldots, t_L, t_{cL}), \qquad t_i \in T_{\text{smi}},\ t_{ci} \in T_{\text{coord}}.$$

Applying the embedding functions yields

$$\boldsymbol{e}_{\text{joint}} = (E_{\text{smi}}(t_1), E_{\text{coord}}(t_{c1}), \ldots, E_{\text{smi}}(t_L), E_{\text{coord}}(t_{cL})) \in \mathbb{R}^{2L\times(d_e+d_t)}.$$

The sequence is then fed to a *single* transformer encoder.

ALT_CONT: Interleaved & Continuous

Here the SMILES tokens remain discrete while the coordinates are kept in their *continuous* form:

$$\boldsymbol{t}_{\text{inC}} = (t_1, \boldsymbol{c_1}, t_2, \boldsymbol{c_2}, \ldots, t_L, \boldsymbol{c_L}), \qquad t_i \in T_{\text{smi}}, \boldsymbol{c_i} \in S_{\text{ics}}.$$

Where S_{ics} is the set of continuous internal coordinates as we defined before. Embeddings are produced as

$$\boldsymbol{e}_{\text{joint}} = (E_{\text{smi}}(t_1), E_{\text{cont}}(\boldsymbol{c}_1), \ldots, E_{\text{smi}}(t_L), E_{\text{cont}}(\boldsymbol{c}_L)) \in \mathbb{R}^{2L\times(d_e+d_t)}.$$

The model architecture is identical to ALT_TOKEN; only the coordinate branch changes.

DE_TOKEN: Dual Encoder & Tokenised

Both streams are tokenised but processed *in parallel*. Separate embedding tables create two length-L sequences

$$\boldsymbol{e}_{\text{smi}} = (E_{\text{smi}}(t_1), \ldots, E_{\text{smi}}(t_L)), \quad \boldsymbol{e}_{\text{coord}} = (E_{\text{coord}}(t_{c1}), \ldots, E_{\text{coord}}(t_{cL})).$$

Two identical transformer encoders encode them independently, after which we concatenate the per-position representations:

$$\boldsymbol{e}_{\text{de}} = (\boldsymbol{e}_{\text{smi}} \| \boldsymbol{e}_{\text{coord}}) \in \mathbb{R}^{L\times 2d_e}.$$

This vector is subsequently projected back to d_e dimensions before being passed to the decoder.

DE_CONT: Dual Encoder & Continuous

Here the SMILES tokens remain discrete while the coordinates are kept in their *continuous* form:

$$\boldsymbol{e}_{\text{smi}} = (E_{\text{smi}}(t_1), \ldots, E_{\text{smi}}(t_L)), \quad \boldsymbol{e}_{\text{coord}} = (E_{\text{cont}}(c_{s1}), \ldots, E_{\text{cont}}(c_{sL})).$$

And then we concatenate them into one embedding vector.

$$\boldsymbol{e}_{\text{de}} = (\boldsymbol{e}_{\text{smi}} \| \boldsymbol{e}_{\text{coord}}) \in \mathbb{R}^{L \times 2d_e}.$$

ALT_TOKEN and ALT_CONT fuse chemistry and geometry early by interleaving at the input layer, differing only in how coordinates are embedded. DE_TOKEN and DE_CONT performs late fusion after two dedicated encoders, allowing each modality to develop its own representation before interaction. All three variants use the same decoder, enabling a controlled comparison of embedding strategies.

3.3.2 Transformer Structure

Here we introduce the transformer structure that is used by all these four models.

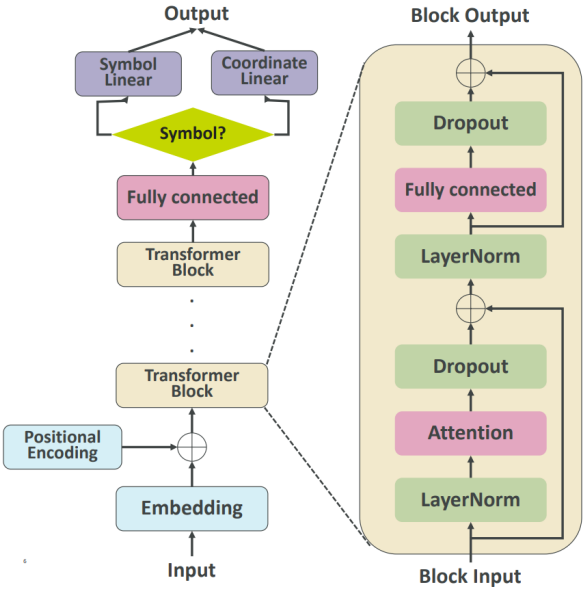


Figure 3.3: Backbone of transformer model. The network couples a joint token-type embedding with sinusoidal position encoding, passes the result through six pre-layer-norm Transformer blocks (8 heads, width depends on models) and branches into two linear heads: one for SMILES symbols and one for coordinate bins.

The encoder comprises $N = 6$ identical transformer blocks, each made of:

1. masked multi-head self-attention with $H = 8$ heads;
2. a feed-forward network of inner width 1024;
3. dropout $p = 0.1$ on both sub-layers.

Padding positions are removed with a key-padding mask, and a causal mask $\mathbf{M}_{ij} = 1_{\{j>i\}}$ blocks information from future positions. After the final layer norm we branch into two linear projections, both heads end in linear and are mutually exclusive.

The generation process follows a simple rule set:

1. SMILES atom → sample a *real* coordinate bin.
2. SMILES non-atom → emit a fixed **placeholder** coordinate token id.
3. Coordinate or Placeholder → sample a SMILES symbol.

Sampling uses temperature scaling and optional top-k truncation. Generation halts on an `<EOS>` emitted from the SMILES head or after a user-supplied length cap.

3.4 Training setup

At decoding step j the network sees the prefix $\boldsymbol{t}^j = (t_0, \ldots, t_j)$ and predicts the next symbol t_{j+1} together with its type (SMILES, coordinate, or placeholder). The joint likelihood of the sequence up to t_{j+1} factorises in the usual autoregressive way:

$$p(\boldsymbol{t}^{j+1}; \theta) \;=\; \prod_{i=0}^{j+1} p\Big(t_i \mid t_{<i}; \theta\Big), \tag{3.1}$$

where θ denotes all network parameters.

Token only variants (ALT_TOKEN, DE_TOKEN). Every position is discrete, so training uses a single cross-entropy objective:

$$\mathcal{L}_{\text{CE}} \;=\; \frac{1}{N_{\text{tok}}} \sum_{\text{batch}} \sum_{j \in \mathcal{T}} \text{CE}\Big(\hat{t}_{j+1},\, t_{j+1}\Big), \tag{3.2}$$

where $\mathcal{T}$ excludes BOS, EOS and PAD tokens.

Mixed variants (ALT_CONT, DE_CONT). Here the model outputs a categorical distribution for SMILES positions and a 3-D vector $\hat{\mathbf{c}}_{j+1} \in S_{ics}$ for coordinates. The loss combines cross-entropy for discrete tokens and mean-squared error for

$$\mathcal{L}_{\text{tot}} \;=\; \frac{1}{N_{\text{tok}}} \sum_{\text{batch}} \sum_{j \in \mathcal{T}} \text{CE}\Big(\hat{t}_{j+1},\, t_{j+1}\Big) \;+\; \frac{\lambda}{N_{\text{cont}}} \sum_{\text{batch}} \sum_{j \in \mathcal{C}} \Big\| \hat{c}_{j+1} - \mathbf{c}_{j+1} \Big\|^2, \tag{3.3}$$

where $\mathcal{C}$ is the index set of coordinate tokens and λ balances the two terms.

Hyper parameters. All values match the launch script used in our experiments.

- **Batch size**: 16
- **Epochs**: 100 with early stopping on validation NLL

- **Optimizer**: Adam, learning rate 3×10^{-4} (constant, no warm-up), $(\beta_1, \beta_2) = (0.9, 0.95)$
- **Drop-out**: $p = 0.1$ on embeddings and Transformer blocks
- **Model width**: token embedding 128, type embedding 16, the total width is 144 for ALT models and 256 for DE models; 8 attention heads, 6 pre-LN layers, FF dimension 1024
- **Hardware**: single NVIDIA A100, 12 data-loader workers

3.5 Autoregressive Generation

The decoder builds a molecule token-by-token, much like writing a sentence. At each step it examines the current prefix and produces two ranked lists of candidates:

- a list of possible **SMILES tokens** (topology), and
- a list of possible **coordinate tokens** (geometry).

Each list is turned into sampling probabilities with a softmax. The temperature T widens or narrows the distribution ($T \uparrow$ more exploratory), and an optional *top-k* filter can keep only the k strongest options.

Unless noted otherwise, we use greedy decoding (temperature T=1, no top-k truncation). Sampling with temperature scaling and optional top-k is also supported.

For the **token types**, we annotate each step with a type label $\tau \in \{0, 1, 2\}$ for SMILES, Coord, and Placeholder, respectively. The label is embedded via a learned lookup and concatenated to the position embedding.

- **If the current token is a SMILES atom** sample a coordinate ID for that atom, mark it as *coordinate* (type 1).
- **If the current token is a SMILES non-atom** (ring digit, branch bracket, etc.) insert a fixed *placeholder* coordinate ID (type 2).
- **If the current token is a coordinate or placeholder** switch back and sample the next SMILES token, mark it as *SMILES* (type 0).

Atom coord, non-atom placeholder, coord/placeholder SMILES. This simple rule keeps the two streams perfectly interleaved and guarantees every heavy atom gets exactly one set of internal coordinates.

Stopping rule

- stop when the decoder emits the end-of-sequence tag, **or**
- stop after reaching the user-defined maximum length.

The final output is a pair of aligned sequences SMILES / placeholder IDs and coordinate IDs-ready to be merged into a full 3-D molecule.

3.6 Evaluation Metrics

A realistic 3-D molecule must clear three hurdles: (i) *chemical* plausibility (valence, charge, syntax), (ii) *physical* plausibility (reasonable energy), and (iii) *geometric* plausibility (near a local minimum). The metrics below target these points from different angles.

- **Validity** Fraction of molecules that pass RDKit sanitisation. Checks valence, charge balance, and SMILES syntax.
- **Novelty** Fraction of valid molecules whose canonical SMILES do not appear in the training set. Gauges the models ability to create new chemistry.
- **Energy validity** Share of molecules for which an MMFF94 single-point energy can be computed. Failure flags severe steric clashes or topologies that break the force field.
- **Energy per atom** MMFF94 energy normalised by atom count. We report both the *mean* (sensitive to outliers) and the *median* (robust to outliers).
- **Strain energy per atom** Difference between the raw and optimised energies, again as mean and median. Large values mean the model places atoms far from equilibrium.
- **Optimised RMSD** Heavy-atom RMSD between the initial and MMFF-optimised conformers. Values below 1 Å indicate near-equilibrium starting geometries.

Table 3.1: Evaluation criteria. Arrows show the desirable direction.

Metric	**What it measures**	**Goal**
Validity	SMILES correctness	↑
Novelty	New chemistry	↑
Energy validity	Force-field compatibility	↑
Mean / median energy/atom	Thermodynamic stability	↓
Mean / median strain/atom	Distance from local minimum	↓
Optimised RMSD	Geometry drift after relaxation	↓

Taken together, these metrics cover symbol, energy, and shape. A model that scores well across the board is likely to generate molecules that are chemically sound, energetically reasonable, and geometrically well behaved.

4

Results

In this chapter we evaluate the four generative models introduced in Chapter 3. Our goal is to determine which representation-embedding combination most reliably produces valid and low energy molecules. Concretely, we contrast discrete vs. continuous internal coordinate tokens, and interleaved vs. concatenated of SMILES and coordinates.

Here we present our generate results.

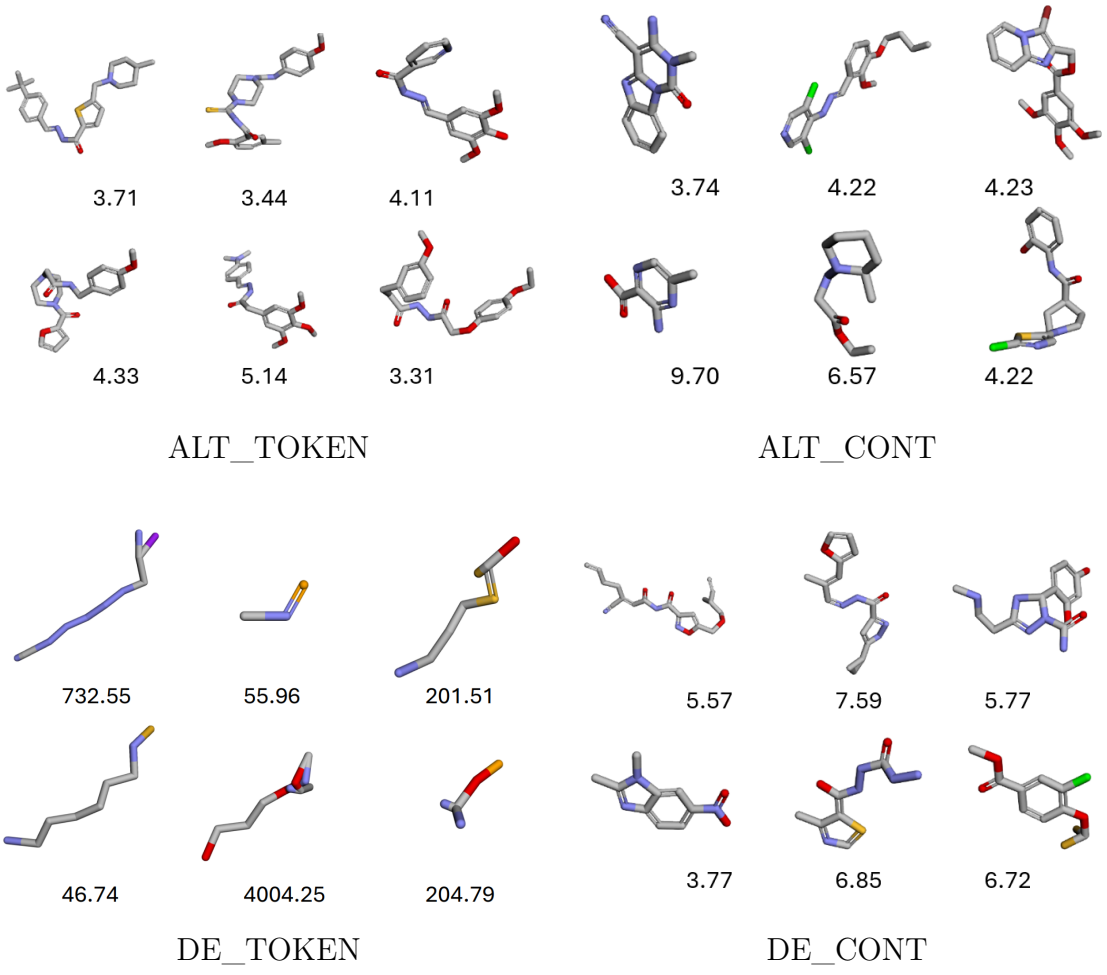


Figure 4.1: Representative conformers drawn from each model (energies printed below each structure).

Figure 4.1 displays typical conformers sampled from each model, with MMFF energies (kcal mol^{-1} atom^{-1}) annotated below each structure. Both **ALT_TOKEN** and **ALT_CONT** generate chemically reasonable, compact geometries with single-digit energies. In contrast, **DE_TOKEN** collapses: most outputs are chemically impossible with high energies. With silghtly high energy the **DE_CONT** variant fares better than its tokenized counterpart and can be compared to the alternating models.

4.1 Quantitative benchmarks

Table 4.1: Performance of the four model variants.

Metric	ALT_TOKEN	ALT_CONT	DE_TOKEN	DE_CONT	DATASET
Validity (%)	**99.0**	87.3	1.4	28.3	
Energy valid (%)	100.0	100.0	100.0	100.0	100.0
Novelty (%)	**100.0**	92.1	**100.0**	98.6	
Avg. energy / atom (kcal mol^{-1})	**171.8**	1 438.1	739.5	3 050.0	1.65
Median energy / atom (kcal mol^{-1})	**3.07**	21.24	2.66	134.40	1.76
Avg. strain energy / atom	**170.3**	1 437.0	739.4	3 047.5	0.32
Median strain energy / atom	**1.02**	19.59	2.56	132.07	0.27
Avg. RMSD (Å)	1.13	**1.09**	1.01	1.52	0.60

Based on the 3D plots presented earlier, it is clear that DE_TOKEN fails to produce meaningful structures. Therefore, we exclude it from all subsequent analyses.

ALT_TOKEN stands out with a 99 % validity rate and perfect novelty, ALT_CONT lowers validity by about twelve percentage points. Both DE variants suffer: DE_CONT retains only moderate validity and DE_TOKEN is essentially unusable. The average MMFF energies are enlarged by a handful of extreme outliers, but the medians reveal a clearer picture. ALT_TOKEN (3.1 kcal mol^{-1} atom^{-1}) almost matches the reference set, whereas DE_CONT remains two orders of magnitude higher. For geometry fidelity ALT_CONT reduce RMSD slightly lower than ALT_TOKEN.

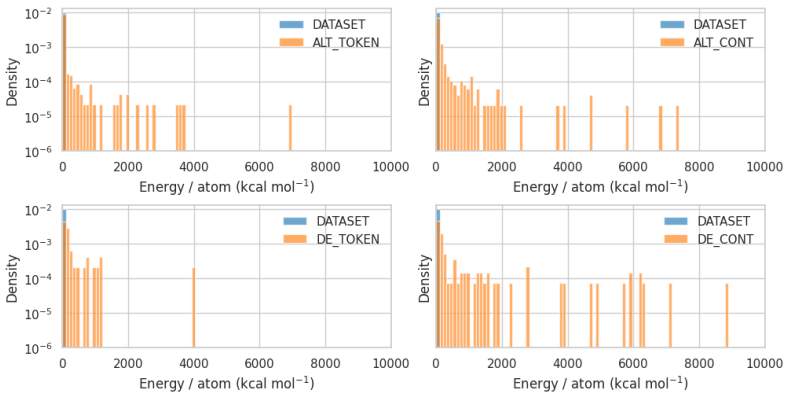


Figure 4.2: Log-density histograms of MMFF94s energy per atom ($\log_{10}$ y-axis). The dataset baseline is sharply peaked (1.76 kcal mol^{-1}). ALT_TOKEN concentrates near the baseline (3.07); ALT_CONT is broader (21.24); DE_CONT exhibits heavy tails (134.40). Medians from Table 4.1 are shown in the legend for reference.

Distributions are shown on a $\log_{10}$ y-axis. Most samples fall below 100 kcal mol^{-1}, while a small fraction of extreme outliers inflates the meanshence we emphasise medians for comparisons. ALT_TOKEN concentrates near the dataset mode; ALT_CONT is broader; DE_CONT shows the heaviest tails.

4.2 Metric distributions

Across four metrics (energy/atom, strain/atom, optimised energy/atom, and RMSD), three patterns emerge (Fig. 4.3).

- ALT_TOKEN tracks the dataset most closely on energy and strain.
- ALT_CONT yields broader energy/strain distributions but attains the lowest post-optimisation RMSD.
- DE_CONT diverges with heavy energy tails.

Together, these suggest that discretised internal coordinates paired with interleaving favour stability and validity, whereas continuous coordinates offer a small RMSD advantage. (We exclude DE_TOKEN from distribution plots due to its failure to produce meaningful structures.)

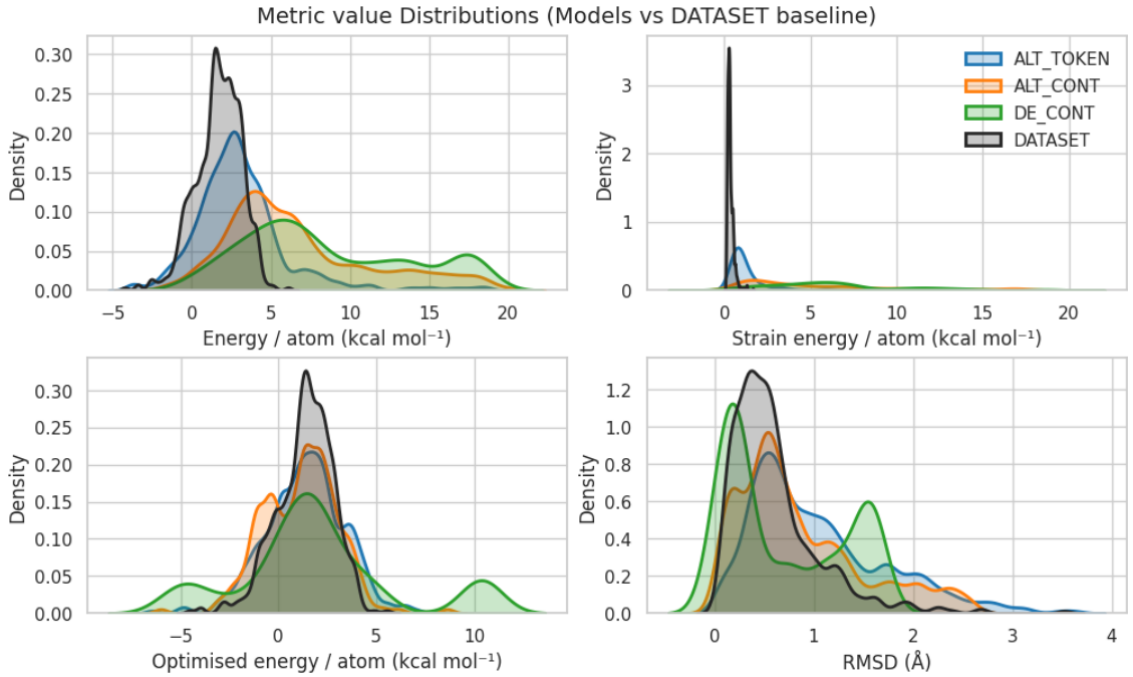


Figure 4.3: Distribution overview across four metrics. Rows: energy/atom, strain energy/atom, optimised energy/atom, and optimised heavy-atom RMSD. Colours: dataset (black), ALT_TOKEN (blue), ALT_CONT (orange), DE_CONT (green). ALT_TOKEN best matches the dataset on energy and strain; ALT_CONT attains the lowest RMSD (1.09 Å vs. 1.13 Å for ALT_TOKEN); DE_CONT shows heavy energy tails. DE_TOKEN is omitted due to invalid outputs.

4.3 Optimisation

This part looks at what happens when we let the MMFF94s optimiser clean up the raw geometries. First, Figure 4.4 shows how the median energy per atom drops over 40 steps for each model and how that compares with the dataset baseline. Next, for each model we pick one *very high-energy* conformer ($>$ 1000 kcal mol^{-1} atom^{-1}). The three chosen molecules all start in the same energy range ($\sim$1700 kcal mol^{-1} atom^{-1}), so they make a fair side-by-side test. We follow each of them through six optimisation snapshots to see which specific geometry errors are responsible for such extreme values.

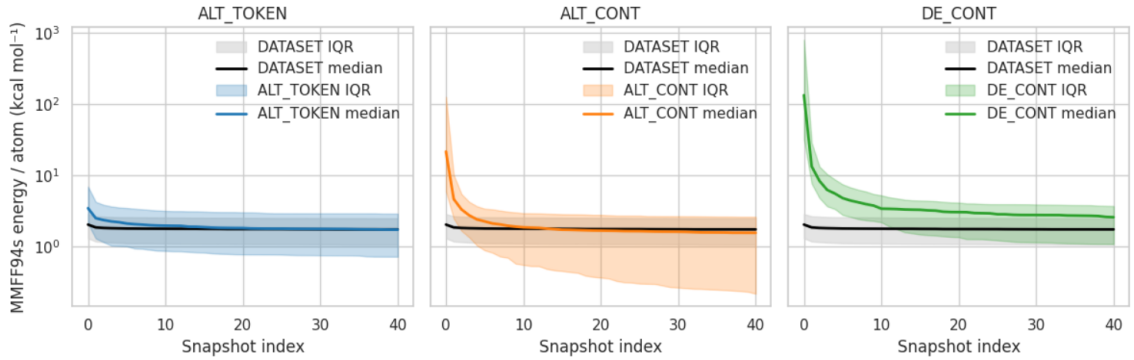


Figure 4.4: Energy-per-atom convergence. Coloured lines are medians; shaded bands are inter-quartile ranges. Black curves are the dataset reference.

Figure 4.4 tracks MMFF94s energy per atom during 40 optimisation steps. ALT_TOKEN (blue) starts a little higher than the dataset, drops fast, and sits on the black baseline after five steps. ALT_CONT (orange) falls even faster and ends slightly *below* the dataset line, showing that continuous coordinates can reach lower energies compare to the reference. DE_CONT (green) begins high and comes down more slowly, the median never meets the baseline.

The next three figures show what makes the worst outliers so expensive.

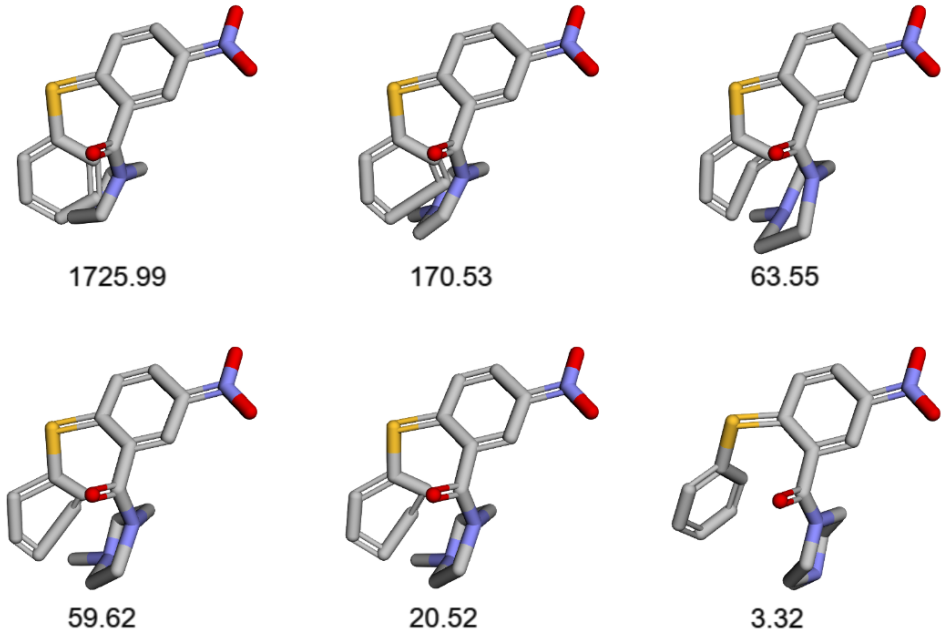


Figure 4.5: ALT_TOKEN outlier. Two rings overlap at the start (1726 kcal mol^{-1} per atom). As the clash is removed the energy falls to 3 kcal mol^{-1}.

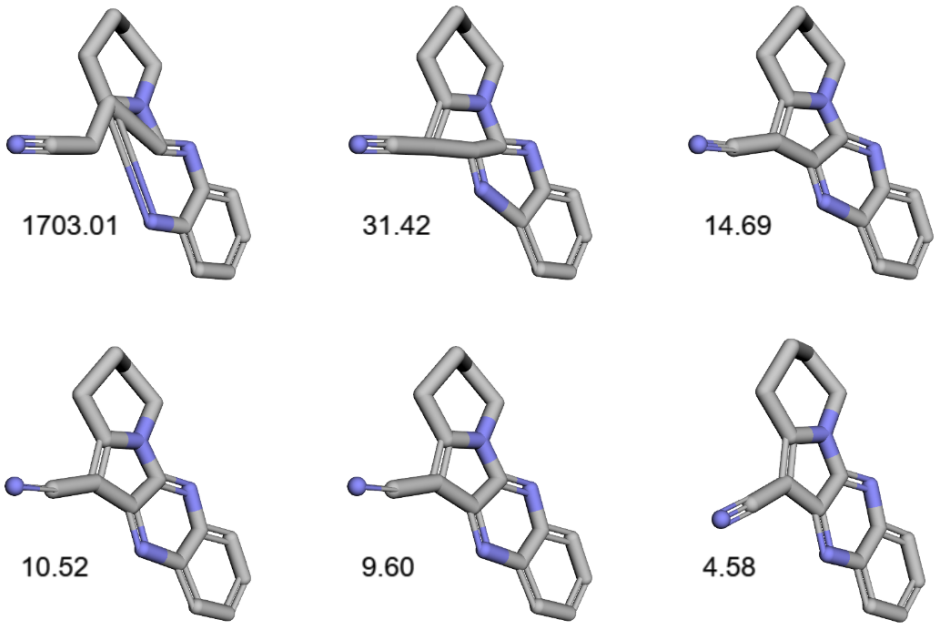


Figure 4.6: ALT_CONT outlier. One dihedral is 0 instead of π, forcing a heavy clash (1703 kcal mol^{-1} per atom). The torsion flips during optimisation and the energy drops below 5 kcal mol^{-1}.

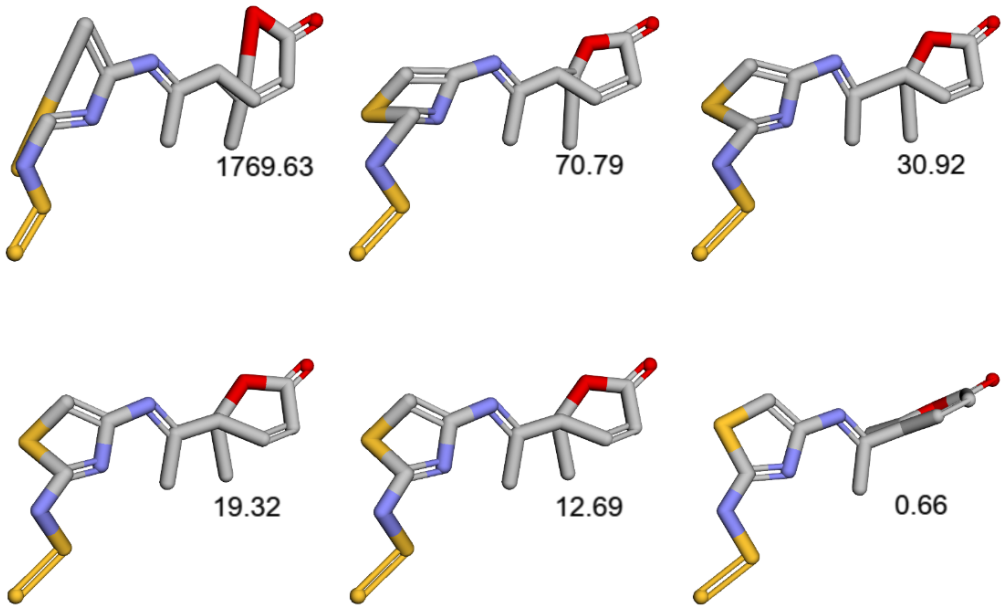


Figure 4.7: DE_CONT outlier. Several bonds are stretched and angles are compressed, so the optimiser can only reduce the energy part-way (from 1770 to about 13 kcal mol^{-1} per atom).

High-energy spikes therefore come from simple geometric mistakes: distorted torsion in ALT_TOKEN, a flipped torsion in ALT_CONT, and many small errors at once in DE_CONT. When the error is local and easy to fix the optimiser recovers quickly; when errors are scattered the molecule stays strained.

After inspecting every conformer with an energy above 1000 kcal mol^{-1} (see Appendix A), we identify three distinct error patterns. ALT_TOKEN outliers are almost always caused by ring overlaps that arise from distorted torsions. ALT_CONT outliers stem mainly from flipped torsions, where a dihedral that should be at 180° is mis-set to 0° (and vice versa). DE_CONT outliers combine several issuesbond stretches, compressed angles, and overlapping side-chains-distributed along the entire backbone, leaving large residual strain after optimisation.

4.4 Discussion

This section addresses the three research questions posed at the outset of the chapter, drawing insights from quantitative benchmarks and qualitative case studies.

RQ1: Minimalist Transformer Viability. Can a standard, non-equivariant Transformer produce chemically valid, accurate, and diverse 3D molecular structures? The results obtained with **ALT_TOKEN** suggest a largely affirmative answer. With a chemical validity of 99%, complete novelty, and a median energy of 3.1 kcal mol^{-1} atom^{-1}, ALT_TOKEN achieves performance comparable to the

reference baseline within statistical uncertainty (see Table 4.1, Figures 4.3, and 4.4). Notably, most initial outliers become chemically valid after only a few optimization steps, indicating that the model inherently generates near-equilibrium geometries. These findings demonstrate that explicit SE(3)-equivariance constraints are not strictly necessary when models are provided with a suitable symmetry-invariant representation.

RQ2: Effectiveness of Alternating Topology and Geometry. Does interleaving SMILES tokens with internal coordinates improve autoregressive 3D generation? Comparing the alternating approaches (ALT_TOKEN, ALT_CONT) with dual-embedding baselines (DE_TOKEN, DE_CONT) reveals clear advantages. While DE_TOKEN largely fails to generate valid structures, and DE_CONT achieves only 28% validity with significantly higher median energies, both alternating variants maintain high validity and produce energy distributions close to the reference. These results strongly indicate that interleaving chemical topology and geometry from the outset maintains consistent synchrony during generation, whereas late fusion (dual embedding) accumulates and amplifies errors.

RQ3: Discrete versus Continuous Internal Coordinates. How do discretized versus continuous internal coordinate representations influence model performance and efficiency? Replacing the discretized coordinate tokens in ALT_TOKEN with continuous floating-point values (ALT_CONT) leads to a 12-percentage-point drop in chemical validity and wider distributions for both energy and RMSD metrics, although it achieves slightly lower median energies after optimization. Continuous coordinates provide more freedom for post-generation optimization but impose higher learning complexity and increase the frequency of suboptimal conformations. Thus, discretization represents a favorable compromise for model convergence and immediate validity, whereas continuous coordinates may be advantageous when achieving absolute lowest energies is crucial and additional optimization steps are acceptable.

Overall, our initial findings suggest that alternating SMILES tokens with discretised, symmetry-invariant internal coordinates within a minimalist Transformer is a particularly promising combination for effective and efficient 3D molecular generation, even without explicit equivariant architectures. While explicit symmetry constraints can clearly offer benefits, it appears that the chosen representation itself plays a critical role in ensuring chemical validity, geometric accuracy, and sampling efficiency. Further studies with comprehensive benchmarks and comparisons would be valuable to fully characterise the potential and limitations of this minimalist, autoregressive approach.

5

Conclusion

5.1 Limitations

- All energies come from MMFF94; higherlevel quantum methods might change the threshold at which geometry failure sets in.
- The coordinate grid is fixed; a learned VQ codebook could offer better compression and smoother interpolation.
- No fragment conditioning, property control or diffusion refinement was explored; these may further improve geometry at a given energy.

5.2 Conclusion

Tokenlevel conditioning with discretised internal coordinates (**ALT_TOKEN**) provides a simple yet effective SE(3)-aware Transformer for 3D molecule generation. On the GEOM-Drugs benchmark it delivers perfect validity, near native energies below 20 kcal mol^{-1}, and realistic conformers, decisively outperforming both a continuous-coordinate variant and a late fusion baseline.

5.3 Future work

The research presented in this thesis successfully demonstrates the viability of using a standard Transformer with an interleaved, discrete representation for 3D molecular generation. The investigation into different combination strategies and representational choices has yielded significant insights. Building upon these findings, several exciting avenues for future work can be identified, aimed at enhancing the model's performance and utility.

5.3.1 Improving Geometric Accuracy

Our results indicate two primary areas for model improvement. First, while our main model, achieved excellent median energy, the high average energy revealed that it occasionally produces physically implausible, high-energy structures. Second, we observed a compelling trade-off between the and variants, where one excels at chemical validity (99.0%) and the other at geometric accuracy (RMSD of 1.01 Å). Future work should therefore aim to create a single, more robust model that unifies

these strengths. This could be pursued by exploring more sophisticated modality fusion strategies, such as gated or cross-attention mechanisms, to create a more synergistic interplay between the topological and geometric information streams. Furthermore, to specifically address the outlier problem, the pre-trained model could be fine-tuned using Reinforcement Learning (RL). By defining a reward function based on low potential energy, this would directly penalize the generation of unstable states and steer the model towards producing an ensemble of molecules that is both chemically valid and possesses high geometric fidelity.

Another option is to develop further the hybrid architecture. While our discrete model offers excellent structural control and validity, the discretization step inherently loses some precision. A similar autoregressive Transformer would be responsible for generating the high-level sequence of topology and coarse geometric tokens. Then, conditioned on the Transformer's hidden state for each coarse geometric token, a small, specialized diffusion model could be used as a decoder head. This diffusion head would be trained to sample a precise, continuous coordinate value from within the bin defined by the coarse token. This hybrid approach could potentially offer the robust, long-range control of an autoregressive model while leveraging the power of diffusion models to generate exceptionally high-fidelity continuous geometries.

5.3.2 Data Representation Strategies

The success of our discrete approach highlights the critical importance of the data representation itself. However, the fixed and uniform nature of our current representation presents clear opportunities for enhancement. A significant next step is to develop an adaptive geometric vocabulary. This could be achieved by pre-training a Vector-Quantized Variational Autoencoder (VQ-VAE) on the dataset's internal coordinate distributions. The resulting learned codebook would provide a more efficient and data-driven discretization that better reflects the true density of chemical space. Currently, the impact of the global atom ordering scheme warrants further investigation. Exploring global orderings derived from alternative, syntactically robust representations like SELFIES, or from graph traversal algorithms, could lead to a more chemically intuitive and more easily learnable sequence for the autoregressive model, potentially improving both chemical and geometric predictions simultaneously.

5.3.3 Controllable and Goal-Directed Generation

To increase the practical utility of this generative framework for real-world drug design campaigns, future efforts should focus on enabling user-controlled, goal-directed generation. The current unconditional model can be extended in two main ways. First, by enabling scaffold-based generation, where the model is conditioned on a user-provided molecular scaffold and tasked with generating novel decorations and side chains. Second, through property conditioning, where target physicochemical properties, such as QED, logP, or predicted binding affinity, are provided as an additional input to the model. This would guide the generation process towards specific, desirable regions of chemical space, transforming our framework from a generative

tool into a goal-oriented molecular design engine.

Bibliography

[1] Y. Cheng, Y. Gong, Y. Liu, B. Song, and Q. Zou, "Molecular design in drug discovery: A comprehensive review of deep generative models," *Briefings in Bioinformatics*, vol. 22, no. 6, bbab344, Aug. 2021, ISSN: 1477-4054. DOI: 10.1093/bib/bbab344.

[2] A. R. von Hippel, "Molecular designing of materials," *Science*, vol. 138, no. 3537, pp. 91–108, 1962. DOI: 10.1126/science.138.3537.91.

[3] Y. Tang, R. Moretti, and J. Meiler, "Recent advances in automated structure-based de novo drug design," *Journal of Chemical Information and Modeling*, vol. 64, no. 6, pp. 1794–1805, 2024.

[4] G. H. Dreiman, M. Bictash, P. V. Fish, L. Griffin, and F. Svensson, "Changing the hts paradigm: Ai-driven iterative screening for hit finding," *SLAS DISCOVERY: Advancing the Science of Drug Discovery*, vol. 26, no. 2, pp. 257–262, 2021.

[5] D. Weininger, "Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules," *Journal of Chemical Information and Computer Sciences*, vol. 28, no. 1, pp. 31–36, 1988.

[6] M. Krenn, F. Häse, A. Nigam, P. Friederich, and A. Aspuru-Guzik, "Self-referencing embedded strings (selfies): A 100% robust molecular string representation," *Machine Learning: Science and Technology*, vol. 1, no. 4, p. 045 024, 2020.

[7] R. Irwin, S. Dimitriadis, J. He, and E. J. Bjerrum, "Chemformer: A pre-trained transformer for computational chemistry," *Machine Learning: Science and Technology*, vol. 3, no. 1, p. 015 022, 2022.

[8] N. C. Frey, R. Soklaski, S. Axelrod, *et al.*, "Neural scaling of deep chemical models," *Nature Machine Intelligence*, vol. 5, no. 11, pp. 1297–1305, 2023.

[9] L. Chanussot, A. Das, S. Goyal, *et al.*, "Open catalyst 2020 (oc20) dataset and community challenges," *Acs Catalysis*, vol. 11, no. 10, pp. 6059–6072, 2021.

[10] S. Axelrod and R. Gómez-Bombarelli, "Geom, energy-annotated molecular conformations for property prediction and molecular generation," *Scientific Data*, vol. 9, no. 1, p. 185, 2022. DOI: 10.1038/s41597-022-01288-4. [Online]. Available: https://doi.org/10.1038/s41597-022-01288-4.

[11] N. W. Gebauer, M. Gastegger, S. S. Hessmann, K.-R. Müller, and K. T. Schütt, "Inverse design of 3d molecular structures with conditional generative neural networks," *Nature communications*, vol. 13, no. 1, p. 973, 2022.

[12] M. Hassan, N. Shenoy, J. Lee, H. Stärk, S. Thaler, and D. Beaini, "Et-flow: Equivariant flow-matching for molecular conformer generation," *Advances in Neural Information Processing Systems*, vol. 37, pp. 128 798–128 824, 2024.

[13] V. Garcia Satorras, E. Hoogeboom, F. Fuchs, I. Posner, and M. Welling, "E (n) equivariant normalizing flows," *Advances in Neural Information Processing Systems*, vol. 34, pp. 4181–4192, 2021.

[14] E. Hoogeboom, V. G. Satorras, C. Vignac, and M. Welling, "Equivariant diffusion for molecule generation in 3d," in *International conference on machine learning*, PMLR, 2022, pp. 8867–8887.

[15] M. Xu, A. S. Powers, R. O. Dror, S. Ermon, and J. Leskovec, "Geometric latent diffusion models for 3d molecule generation," in *International Conference on Machine Learning*, PMLR, 2023, pp. 38 592–38 610.

[16] R. Irwin, A. Tibo, J. P. Janet, and S. Olsson, "Semlaflow–efficient 3d molecular generation with latent attention and equivariant flow matching," in *The 28th International Conference on Artificial Intelligence and Statistics*, 2025.

[17] M. Buttenschoen, G. M. Morris, and C. M. Deane, "Posebusters: Ai-based docking methods fail to generate physically valid poses or generalise to novel sequences," *Chemical Science*, vol. 15, no. 9, pp. 3130–3139, 2024.

[18] C. Harris, K. Didi, A. R. Jamasb, *et al.*, "Posecheck: Generative models for 3d structure-based drug design produce unrealistic poses," 2023.

[19] J. Abramson, J. Adler, J. Dunger, *et al.*, "Accurate structure prediction of biomolecular interactions with alphafold 3," *Nature*, vol. 630, no. 8016, pp. 493–500, 2024.

[20] T. Geffner, K. Didi, Z. Zhang, *et al.*, "Proteina: Scaling flow-based protein structure generative models," *arXiv preprint arXiv:2503.00710*, 2025.

[21] Y. Wang, A. A. Elhag, N. Jaitly, J. M. Susskind, and M. A. Bautista, "Swallowing the bitter pill: Simplified scalable conformer generation," *arXiv preprint arXiv:2311.17932*, 2023.

[22] Y. Song, J. Gong, M. Xu, *et al.*, "Equivariant flow matching with hybrid probability transport for 3d molecule generation," *Advances in Neural Information Processing Systems*, vol. 36, pp. 549–568, 2023.

[23] C. K. Joshi, X. Fu, Y.-L. Liao, *et al.*, "All-atom diffusion transformers: Unified generative modelling of molecules and materials," *arXiv preprint arXiv:2503.03965*, 2025.

[24] Y. Luo and S. Ji, "An autoregressive flow model for 3d molecular geometry generation from scratch," in *International conference on learning representations (ICLR)*, 2022.

[25] A. Daigavane, S. Kim, M. Geiger, and T. Smidt, "Symphony: Symmetry-equivariant point-centered spherical harmonics for 3d molecule generation," *arXiv preprint arXiv:2311.16199*, 2023.

[26] K. Gao, Y. Wang, H. Guan, *et al.*, "Tokenizing 3d molecule structure with quantized spherical coordinates," *arXiv preprint arXiv:2412.01564*, 2024.

[27] A. H. Cheng, C. Sun, and A. Aspuru-Guzik, "Scalable autoregressive 3d molecule generation," *arXiv preprint arXiv:2505.13791*, 2025.

[28] T. Li, Y. Tian, H. Li, M. Deng, and K. He, "Autoregressive image generation without vector quantization," *Advances in Neural Information Processing Systems*, vol. 37, pp. 56 424–56 445, 2024.

[29] N. O'Boyle and A. Dalke, "Deepsmiles: An adaptation of smiles for use in machine-learning of chemical structures," 2018.

[30] D. Weininger, A. Weininger, and J. L. Weininger, "Smiles. 2. algorithm for generation of unique smiles notation," *Journal of chemical information and computer sciences*, vol. 29, no. 2, pp. 97–101, 1989.

[31] J. Lim, S. Ryu, K. Park, Y. J. Choe, J. Ham, and W. Y. Kim, "Predicting drug–target interaction using a novel graph neural network with 3d structure-embedded graph representation," *Journal of chemical information and modeling*, vol. 59, no. 9, pp. 3981–3988, 2019.

[32] M. Geiger and T. Smidt, "E3nn: Euclidean neural networks," *arXiv preprint arXiv:2207.09453*, 2022.

[33] F. Jensen, *Introduction to computational chemistry*. John wiley & sons, 2017.

[34] A. R. Leach, *Molecular modelling: principles and applications*. Pearson education, 2001.

[35] D. Young, *Computational Chemistry: A Practical Guide for Applying Techniques to Real World Problems*. John Wiley & Sons, 2004, pp. 73–77.

[36] J. Parsons, J. B. Holmes, J. M. Rojas, J. Tsai, and C. E. Strauss, "Practical conversion from torsion space to cartesian space for in silico protein synthesis," *Journal of computational chemistry*, vol. 26, no. 10, pp. 1063–1068, 2005.

[37] T. A. Halgren, "Merck Molecular Force Field. I. Basis, Form, Scope, Parameterization, and Performance of MMFF94," *Journal of Computational Chemistry*, vol. 17, no. 5–6, pp. 490–519, 1996. DOI: 10.1002/(SICI)1096-987X(199604)17:5/6<490::AID-JCC1>3.0.CO;2-P.

[38] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Grids, groups, graphs, geodesics, and gauges," *arXiv preprint arXiv:2104.13478*, 2021.

[39] J. Köhler, L. Klein, and F. Noé, "Equivariant flows: Exact likelihood generative learning for symmetric densities," in *International conference on machine learning*, PMLR, 2020, pp. 5361–5370.

[40] V. G. Satorras, E. Hoogeboom, and M. Welling, "E (n) equivariant graph neural networks," in *International conference on machine learning*, PMLR, 2021, pp. 9323–9332.

[41] N. Thomas, T. Smidt, S. Kearnes, *et al.*, "Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds," *arXiv preprint arXiv:1802.08219*, 2018.

[42] F. Fuchs, D. Worrall, V. Fischer, and M. Welling, "Se (3)-transformers: 3d roto-translation equivariant attention networks," *Advances in neural information processing systems*, vol. 33, pp. 1970–1981, 2020.

[43] M. Kuznetsov, F. Ryabov, R. Schutski, *et al.*, "Cosmic: Molecular conformation space modeling in internal coordinates with an adversarial framework," *Journal of Chemical Information and Modeling*, vol. 64, no. 9, pp. 3610–3620, 2024.

[44] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[46] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[47] D. Reidenbach, F. Nikitin, O. Isayev, and S. Paliwal, "Applications of modular co-design for de novo 3d molecule generation," *arXiv preprint arXiv:2505.18392*, 2025.

[48] M. Buttenschoen, Y. Ziv, G. M. Morris, and C. M. Deane, "An evaluation of unconditional 3d molecular generation methods," *arXiv preprint arXiv:2505.00518*, 2025.

[49] I. Dunn and D. R. Koes, "Mixed continuous and categorical flow matching for 3d de novo molecule generation," *ArXiv*, arXiv–2404, 2024.

[50] Z. Liu, Y. Luo, H. Huang, *et al.*, "Next-mol: 3d diffusion meets 1d language modeling for 3d molecule generation," *arXiv preprint arXiv:2502.12638*, 2025.

[51] Y. Ding and T. Hofmann, "Scalable non-equivariant 3d molecule generation via rotational alignment," *arXiv preprint arXiv:2506.10186*, 2025.

[52] Anonymous, *A standard transformer and attention with linear biases for molecular conformer generation*, 2025. [Online]. Available: `https://openreview.net/forum?id=BjjerMYL3F`.

[53] M. Arts, J. Frellsen, and W. Boomsma, "Internal-coordinate density modelling of protein structure: Covariance matters," *arXiv preprint arXiv:2302.13711*, 2023.

[54] A. Gaulton, L. J. Bellis, A. P. Bento, *et al.*, "Chembl: A large-scale bioactivity database for drug discovery," *Nucleic acids research*, vol. 40, no. D1, pp. D1100–D1107, 2012.

[55] J. J. Irwin and B. K. Shoichet, "Zinc- a free database of commercially available compounds for virtual screening," *Journal of chemical information and modeling*, vol. 45, no. 1, pp. 177–182, 2005.

[56] U. Norinder, P. Lidén, and H. Boström, "Discrimination between modes of toxic action of phenols using rule based methods," *Molecular diversity*, vol. 10, pp. 207–212, 2006.

[57] F. Nikitin, I. Dunn, D. R. Koes, and O. Isayev, "Geom-drugs revisited: Toward more chemically accurate benchmarks for 3d molecule generation," *arXiv preprint arXiv:2505.00169*, 2025.

[58] H. H. Loeffler, J. He, A. Tibo, *et al.*, "Reinvent 4: Modern ai–driven generative molecule design," *Journal of Cheminformatics*, vol. 16, no. 1, p. 20, 2024.

A

Appendix 1

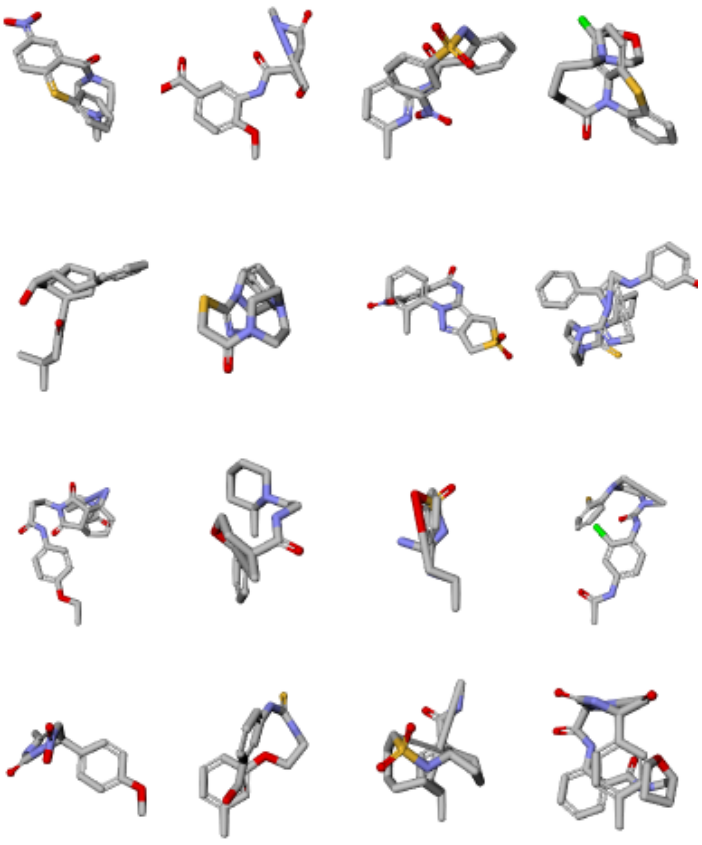


Figure A.1: ALT_TOKEN: molecule energy less than 1000 kcal/mol per atom.

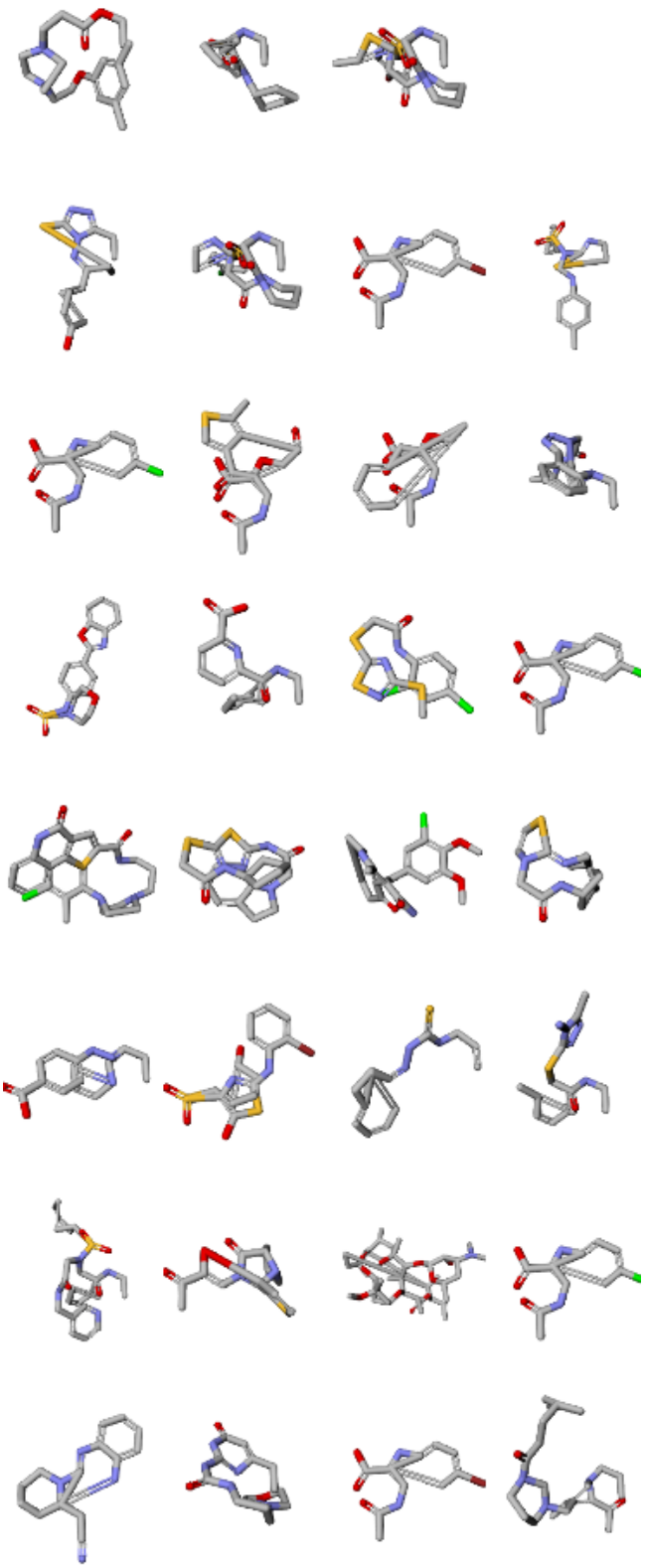


Figure A.2: ALT_CONT: molecule energy less than 1000 kcal/mol per atom.

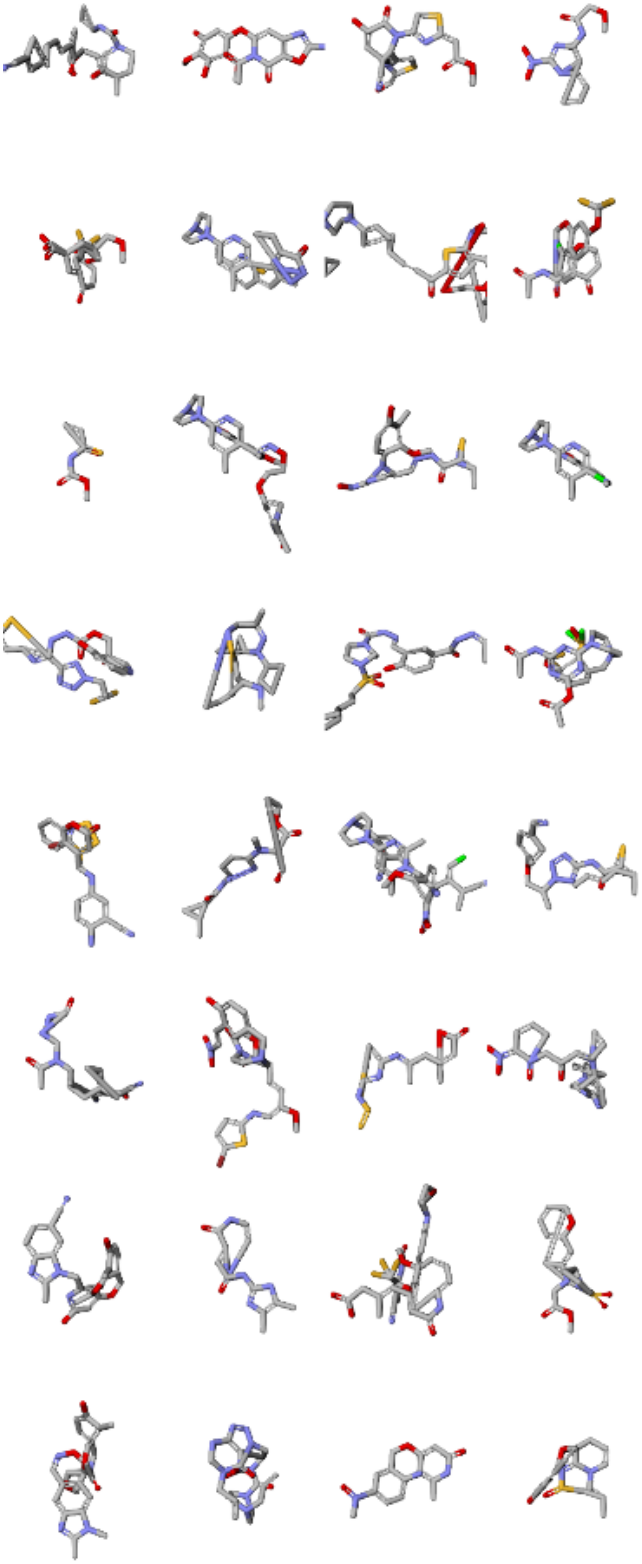


Figure A.3: DE_CONT: molecule energy less than 1000 kcal/mol per atom.