SemlaFlow – Efficient 3D Molecular Generation with Latent Attention and Equivariant Flow Matching

Ross Irwin[1,2*] **Alessandro Tibo**[1] **Jon Paul Janet**[1] **Simon Olsson**[2]

[1]Molecular AI
Discovery Sciences, R&D
AstraZeneca
Gothenburg, Sweden

[2]Department of Computer Science and Engineering
Chalmers University of Technology
and University of Gothenburg
Gothenburg, Sweden

Abstract

Methods for jointly generating molecular graphs along with their 3D conformations have gained prominence recently due to their potential impact on structure-based drug design. Current approaches, however, often suffer from very slow sampling times or generate molecules with poor chemical validity. Addressing these limitations, we propose Semla, a scalable E(3)-equivariant message passing architecture. We further introduce an unconditional 3D molecular generation model, SemlaFlow, which is trained using equivariant flow matching to generate a joint distribution over atom types, coordinates, bond types and formal charges. Our model produces state-of-the-art results on benchmark datasets with as few as 20 sampling steps, corresponding to a two order-of-magnitude speedup compared to state-of-the-art. Furthermore, we highlight limitations of current evaluation methods for 3D generation and propose new benchmark metrics for unconditional molecular generators. Finally, using these new metrics, we compare our model's ability to generate high quality samples against current approaches and further demonstrate SemlaFlow's strong performance.

1 INTRODUCTION

Generative models for 3D drug design have recently seen a surge of interest due to their potential to design binders for given protein pockets. Some recently proposed models have attempted to directly generate ligands within rigid pockets (Peng et al., 2022; Guan et al., 2023; Schneuing et al., 2022). More thorough analysis, however, revealed that many of these models generate ligands with unrealistic binding poses (Buttenschoen et al., 2024; Harris et al., 2023). Others have attempted to train unconditional 3D molecular generators as a starting point (Hoogeboom et al., 2022; Song et al., 2023; Vignac et al., 2023; Hua et al., 2024; Morehead and Cheng, 2024; Xu et al., 2024; Le et al., 2024). Specifically, models which apply diffusion (Ho et al., 2020; Song et al., 2021) to molecular coordinates have been particular popular. However, these models also suffer significant practical limitations; namely, they almost all require hundreds or even thousands of forward passes during generation, making them impractical for most downstream applications. Many also generate chemically unrealistic or poor quality samples when applied to datasets of drug-like molecules.

For molecular generators which represent molecules as strings or 2D graphs, fine-tuning for specific protein pockets has proven very fruitful (Blaschke et al., 2020; Loeffler et al., 2024; Atance et al., 2022) and is currently standard practice in the field. Frequently, these models are guided into optimised chemical spaces using reinforcement learning (RL). This approach, while very effective, requires that high quality molecules can be sampled very quickly. Existing 3D molecular generators, which use fully-connected message passing, exhibit poor scaling to larger molecules and larger model sizes – state-of-the-art unconditional generators (Vignac et al., 2023; Le et al., 2024) take minutes to sample a single batch, making them impractical for RL-based fine-tuning.

In this work we tackle this problem from two directions. Firstly, we introduce a novel equivariant architecture

Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258.

*Correspondence to rossir@chalmers.se

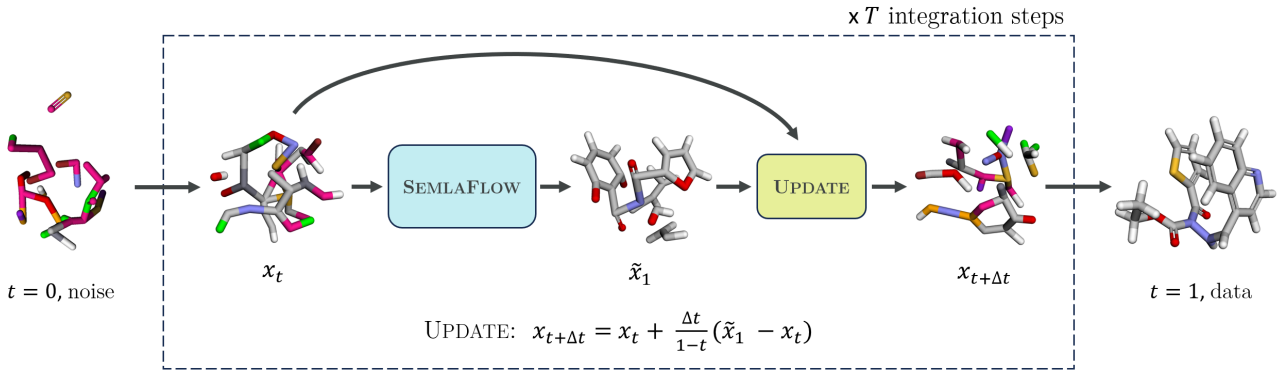

Figure 1: An overview of the inference procedure for SemlaFlow. Noise is firstly sampled from the prior distribution over atom types, bond types and coordinates. We then integrate the vector field by applying our current prediction to the SemlaFlow model and taking a step in the direction of the model's prediction. To improve readability noisy samples are displayed with bonds inferred based on the 3D coordinates.

for 3D graph generation, Semla, where attention between nodes is applied in a reduced latent space. Our architecture exhibits significantly better efficiency and scalability than existing approaches; even with twice as many parameters as the current state-of-the-art, our model processes a batch of molecules more than three times as quickly. Secondly, we propose a flow matching model for 3D molecular generation which learns a joint distribution over the molecular topology, atomic coordinates and formal charges. Our model, SemlaFlow, provides state-of-the-art results with as few as 20 sampling steps, corresponding to a 2-order-of-magnitude increase in sampling speed compared to existing models. Finally, we also highlight issues with frequently used evaluation metrics for unconditional 3D generation and propose the use of energy and strain energy for benchmarking unconditional 3D generative models.

We provide an overview of our molecular generation approach in Figure 1 and we summarise our key contributions as follows:

- **Semla** – We introduce a novel E(3)-equivariant architecture, Semla, with significantly better scalability and efficiency than previous molecular generation approaches.
- **SemlaFlow** – A state-of-the-art molecular generator trained using flow matching with equivariant optimal transport, which provides a more than 100-fold improvement in sampling time compared to existing approaches while maintaining state-of-the-art performance.
- **Benchmarking** – We introduce new evaluation metrics to address a number of shortcomings with existing metrics for 3D molecular generation.

2 BACKGROUND

Flow Matching Flow matching seeks to learn a generative process which transports samples from a noise distribution p_{noise} to samples from a data distribution p_{data}. Conditional flow matching (CFM) has emerged in different flavors as an effective way to train flow matching models (Albergo and Vanden-Eijnden, 2023; Liu et al., 2023; Lipman et al., 2023). CFM works in a simulation-free manner by interpolating between noise and a data sample $x_1 \sim p_{\text{data}}(\mathbf{x}_1)$. To do this a time-dependent conditional flow $p_{t|1}(\cdot|\mathbf{x}_1)$ is defined, from which a conditional vector field $u_t(\mathbf{x}_t|\mathbf{x}_1)$ can be derived. Typically, a model $v_t^\theta(\mathbf{x}_t)$ is trained to regress the vector field, but other formulations (Campbell et al., 2024; Stark et al., 2024a) have trained a model to estimate the distribution $p_{1|t}^\theta(\cdot|\mathbf{x}_t)$, which reconstructs clean data from noisy data. The vector field can then be constructed using the expectation:

$$v_t^\theta(\mathbf{x}_t) = \mathbb{E}_{\tilde{\mathbf{x}}_1 \sim p_{1|t}^\theta(\mathbf{x}_1|\mathbf{x}_t)}([u_t(\mathbf{x}_t|\tilde{\mathbf{x}}_1)]) \tag{1}$$

Samples can then be generated by integrating the vector field with an arbitrary ODE solver.

Invariance and Equivariance Group invariance and equivariance are crucial properties to consider when designing models for 3D molecular generation. For a group $\mathcal{G}$, if T_g and P_g are linear representations of a group element $g \in \mathcal{G}$, then a probability density $p(\mathbf{x})$ is considered *invariant* with respect to $\mathcal{G}$ iff $p(T_g\mathbf{x}) = p(\mathbf{x})$ for all $g \in \mathcal{G}$, and a function f is considered *equivariant* to $\mathcal{G}$ iff $T_g(f(\mathbf{x})) = f(P_g(\mathbf{x}))$ for all $g \in \mathcal{G}$.

Köhler et al. (2020) showed that, if a base density $p_0(\mathbf{x}_0)$ is $\mathcal{G}$-invariant and a target density $p_1(\mathbf{x}_1)$ is generated by following a $\mathcal{G}$-equivariant vector field, then $p_1(\mathbf{x}_1)$ is also $\mathcal{G}$-invariant. We use this finding

to ensure that the density of molecular coordinates learned by our model is $\mathcal{G}$-invariant by only applying equivariant updates and sampling coordinate noise from an isotropic Gaussian. For molecular generation we are concerned with the group $\mathcal{G} = \mathrm{E}(3) \times S_N$ where E(3) is the Euclidean group in 3 dimensions, encompassing translations, rotations and reflections, and S_N is the symmetric group for a set with N elements – the group of all possible permutations.

3 The Semla Architecture

Existing state-of-the-art models for 3D molecular generation (Hoogeboom et al., 2022; Vignac et al., 2023; Le et al., 2024) use fully-connected, multi-layer perceptron (MLP)-based message passing layers. However, the computational cost of such layers scales quadratically in both the feature dimension and the number of atoms. Consequently, these layers become a significant computational bottleneck when scaling to larger, drug-like molecules.

To alleviate this problem we propose Semla - a scalable equivariant model which uses multi-head latent graph attention, where message passing is performed on compressed latent representations. This extension allows us to scale the dimensionality of the node features and the number of learnable model parameters without leading to prohibitory increases in computational cost. We illustrate one Semla architecture in Figure 2 and further expand on each component below.

Similarly to previous approaches, Semla makes use of both E(3) invariant and equivariant features. Enforcing group symmetry provides strong inductive biases and improves sample complexity (Bietti et al., 2021; Tahmasebi and Jegelka, 2024; Hoogeboom et al., 2022). However, unlike many existing molecular generation architectures, Semla does not distinguish between molecular coordinates and equivariant feature vectors, but rather treats them as a single learnable representation. We argue that this representation has two key benefits over previous molecular generation approaches. Firstly, sets of learnable equivariant features provide much more expressivity than models which store only one set of coordinates per molecule (Satorras et al., 2021; Hoogeboom et al., 2022; Vignac et al., 2023). Additionally, a joint representation of equivariant features allows for a simpler update mechanism than that proposed in EQGAT (Le et al., 2022) – we can simply apply linear projections (without bias) to create and update the feature vectors while maintaining equivariance.

To ensure stable training we use normalisation layers throughout the model. LayerNorm (Ba et al., 2016) is applied to invariant features, and, for equivariant features, we adapt the normalisation scheme

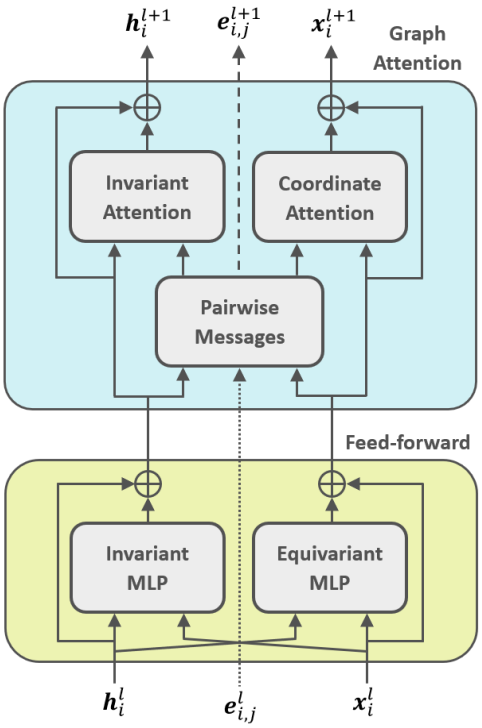

Figure 2: Architectural overview of one Semla layer.

from MiDi (Vignac et al., 2023) to allow for multiple equivariant feature vectors. We hypothesise that this allows the model to learn equivariant features of different length scales, which helps to circumvent the problem of molecules of different sizes being normalised to have the same average vector norm. We further extend the normalisation to ensure that coordinates are zero-centred. An additional zero-centering, which we apply at the end of the model, ensures that the learned density is translation invariant (Garcia Satorras et al., 2021; Xu et al., 2022). For the remainder of this paper we will use $\phi_{inv}(\cdot)$ and $\phi_{equi}(\cdot)$ to refer to the normalisation functions for invariant and equivariant features, respectively. Both LayerNorm and our coordinate normalisation module contain a small number of learnable parameters, these are not shared between layers.

We will denote invariant and equivariant features for an atom i as $\mathbf{h}_i \in \mathbb{R}^{d_{inv}}$ and $\mathbf{x}_i \in \mathbb{R}^{d_{equi} \times 3}$, respectively, where d_{inv} is the number of invariant scalar features and d_{equi} is the number of equivariant feature vectors. We also use N to refer to the number of atoms in the molecule. To simplify the notation, we assume that operations applied to $\mathbf{x}_i$ implicitly correspond to the concatenation of the results of the operation applied to individual vectors, unless we make the equivariant feature vector explicit using a superscript. For example, the norm of $\mathbf{x}_i$ is implicitly applied as $\|\mathbf{x}_i\| = \left[\|\mathbf{x}_i^1\|, \|\mathbf{x}_i^2\|, \ldots, \|\mathbf{x}_i^{d_{equi}}\|\right]$.

3.1 Feature Feed-forward

The feed-forward component provides a simple feature update mechanism while also allowing the exchange of information between invariant and equivariant features. The feed-forward update is given as follows:

$$\tilde{\mathbf{h}}_k = \phi_{inv}(\mathbf{h}_k) \qquad \tilde{\mathbf{x}}_k = \mathbf{W}^1_\theta \phi_{equi}(\mathbf{x}_k) \tag{2}$$

$$\mathbf{h}^{\text{ff}}_i = \mathbf{h}_i + \Phi_\theta(\tilde{\mathbf{h}}_i, \|\phi_{equi}(\mathbf{x}_i)\|) \tag{3}$$

$$\mathbf{x}^{\text{ff}}_i = \mathbf{x}_i + \mathbf{W}^2_\theta \left(\sum_{j=1}^{d_{equi}} \tilde{\mathbf{x}}^j_i \otimes \Psi_\theta(\tilde{\mathbf{h}}_i) \right) \tag{4}$$

where Φ_θ and Ψ_θ are learnable multi-layer perceptrons, $\mathbf{W}^1_\theta \in \mathbb{R}^{d_{equi} \times d_{equi}}$ and $\mathbf{W}^2_\theta \in \mathbb{R}^{d_{equi} \times d_{equi}}$ are learnable weight matrices, and $\otimes$ is the outer product. Similarly to the transformer architecture (Vaswani et al., 2017), Φ_θ linearly maps features to $4 \times d_{inv}$, applies a non-linearity (we use SiLU (Elfwing et al., 2018) throughout) and then maps back to d_{inv}.

3.2 Equivariant Graph Attention

In this section, we introduce a novel attention mechanism for 3D graph structures. Like previously proposed attention mechanisms for equivariant architectures (Satorras et al., 2021; Le et al., 2022; Liao and Smidt, 2023), our model computes pairwise messages using a multi-layer perceptron (MLP). These pairwise messages are then split in two and passed to attention mechanisms for invariant and equivariant features, respectively. We describe each of these components in more detail below.

Latent Message Passing Pairwise messages are computed using a 2-layer MLP which combines invariant node features with pairwise dot products from the equivariant features. Similarly to previous approaches we compute messages between all pairs of nodes in the graph. Unlike models such as EGNN (Satorras et al., 2021), MiDi (Vignac et al., 2023) and EQGAT (Le et al., 2022), however, we first compress the invariant node features into a smaller latent space, with dimensionality d_l, using a learnable linear map. This reduces the computational complexity of the pairwise MLP from $\mathcal{O}(N^2 d^2_{inv})$ to $\mathcal{O}(N^2 d^2_l)$ where $d_l \ll d_{inv}$, leading to a significant reduction in the compute and memory overhead of the MLP, especially on larger molecules. It also allows us to scale the size of the invariant node features independently of the node latent dimension. In Appendix A we provide experiments varying the size of this parameter and show that it is possible to produce a significant reduction in inference time with negligible drop in generation quality. Formally, messages between nodes i and j, which are split into invariant and equivariant attention components, are computed as follows:

$$(\mathbf{m}^{(inv)}_{i,j}, \mathbf{m}^{(equi)}_{i,j}) = \Omega_\theta(\tilde{\mathbf{h}}_i, \tilde{\mathbf{h}}_j, \tilde{\mathbf{x}}_i \cdot \tilde{\mathbf{x}}_j) \tag{5}$$

$$\tilde{\mathbf{h}}_k = \mathbf{W}^3_\theta \phi_{inv}(\mathbf{h}^{\text{ff}}_k) \qquad \tilde{\mathbf{x}}_k = \phi_{equi}(\mathbf{x}^{\text{ff}}_k) \tag{6}$$

where Ω_θ is the pairwise message MLP and $\mathbf{W}^3_\theta \in \mathbb{R}^{d_l \times d_{inv}}$ is a learnable latent projection matrix.

Invariant Feature Attention Once messages have been computed a softmax operation is applied to produce attention weights between pairs of nodes. These weights are then used to aggregate node features by taking a weighted average. Since the message vectors can, in general, be smaller than the node features, each scalar in the message vector attends to a fixed number of scalars within the node feature vectors. We note that this attention implementation generalises the attention mechanism found in EQGAT and related models such as the Point Transformer (Zhao et al., 2021), where each scalar in the message attends to exactly one scalar in the node features, and is very closely related to the multi-head attention mechanism adopted in GAT (Veličković et al., 2018; Brody et al., 2021) and the Transformer (Vaswani et al., 2017). We also make use of the recently proposed variance preserving aggregation mechanism (Schneckenreiter et al., 2024), which corresponds to multiplying the attended vectors by weights w^k_i. Overall, our invariant feature attention is computed as:

$$\alpha^k_{i,j} = \frac{\exp(m^{k,(inv)}_{i,j})}{\sum_{j'=1}^N \exp(m^{k,(inv)}_{i,j'})} \qquad w^k_i = \sqrt{\sum_{j=1}^N (\alpha^k_{i,j})^2} \tag{7}$$

$$\tilde{\mathbf{h}}_i = \mathbf{W}^4_\theta \phi_{inv}(\mathbf{h}^{\text{ff}}_i) \qquad \mathbf{a}^k_i = \sum_{j=1}^N \alpha^k_{i,j} \tilde{\mathbf{h}}^k_j \tag{8}$$

$$\mathbf{h}^{\text{out}}_i = \mathbf{h}^{\text{ff}}_i + \mathbf{W}^5_\theta \Big(\Big\|_{k=1}^K w^k_i \mathbf{a}^k_i \Big) \tag{9}$$

where $\mathbf{W}^4_\theta \in \mathbb{R}^{d_{inv} \times d_{inv}}$ and $\mathbf{W}^5_\theta \in \mathbb{R}^{d_{inv} \times d_{inv}}$ are learnable weight matrices and $\|$ is the concatenation operation. Here, node features are split into n_{heads} equally sized segments and each scalar attention score $\alpha^k_{i,j}$ attends to one segment $\tilde{\mathbf{h}}^k_j$.

Equivariant Feature Attention Similarly to the invariant features, messages for the equivariant features are used to apply an attention-based update. The attention function applied here is line with previous work (Satorras et al., 2021; Vignac et al., 2023; Le et al., 2022), however we extend it to allow for multiple equivariant feature vectors. Notably, we also find the use of softmax normalisation on raw messages to be beneficial for overall model performance. Analogously to the invariant feature attention, we also apply variance preserving updates to the equivariant features.

SEMLA attention for equivariant features is therefore defined as follows:

$$\hat{\mathbf{x}}_{i,j} = \frac{\tilde{\mathbf{x}}_j - \tilde{\mathbf{x}}_i}{\|\tilde{\mathbf{x}}_j - \tilde{\mathbf{x}}_i\|} \qquad \tilde{\mathbf{x}}_k = \mathbf{W}_\theta^6 \phi_{equi}(\mathbf{x}_k^{\text{ff}}) \tag{10}$$

$$\alpha_{i,j}^k = \frac{\exp(m_{i,j}^{k,(equi)})}{\sum_{j'=1}^N \exp(m_{i,j'}^{k,(equi)})} \tag{11}$$

$$a_i^k = \sum_{j=1}^N \alpha_{i,j}^k \hat{x}_{i,j}^k \qquad w_i^k = \sqrt{\sum_{j=1}^N (\alpha_{i,j}^k)^2} \tag{12}$$

$$\mathbf{x}_k^{\text{out}} = \mathbf{x}_k^{\text{ff}} + \mathbf{W}_\theta^7 \Big(\left[w_i^1 a_i^1, \ldots, w_i^K a_i^K \right]^T \Big) \tag{13}$$

3.3 Overall Architecture

A full SEMLA model consists of a stack of SEMLA layers along with embedding layers and MLPs for encoding the atom and bond types, and MLP prediction heads for producing unnormalised distributions for atoms, bonds and formal charges. Unlike EQGAT, our model does not carry edge features throughout the network. Instead, the first layer embeds bond information into the node features by passing the encoded bond features into the pairwise message module. Analogously, the final layer produces pairwise edge features which are then further updated through a bond refinement layer at the end of the network. This layer acts in a similar way to the pairwise message block described above but only updates the edge features. Absorbing bond information into the node features like this leads to a further increase in the efficiency of our model and, in our experiments, had little impact on generative performance.

Unless stated otherwise, the SEMLA models we present in the remainder of this paper are constructed from 12 layers with $d_{inv} = 384$, $d_{equi} = 64$, $d_l = 64$, $n_{heads} = 32$. This corresponds to approximately 22M learnable parameters. We provide a full model overview and further hyperparameter details in Appendix C.

4 FLOW MATCHING FOR MOLECULAR GENERATION

To assess the ability of SEMLA to model distributions with $E(3) \times S_N$ symmetry we apply conditional flow matching (Lipman et al., 2023; Albergo and Vanden-Eijnden, 2023; Albergo et al., 2023; Liu et al., 2023) with optimal transport to create a generative model for molecules, which we refer to as SEMLAFLOW. Unlike many previous approaches to 3D generation, we learn to generate a join distribution over atomic coordinates, atom types, bond orders and formal charges, rather than inferring parts of this distribution after generation. Each molecule is represented by a tuple $z = (\mathbf{x}, \mathbf{a}, \mathbf{b}, \mathbf{c})$ of coordinates, atom types, bond orders and formal charges, respectively. Since we wish to train a model to sample from the joint distribution $p(z)$ which contains both discrete and continuous data, we parameterise a single SEMLA model to generate multiple vector fields. For discrete data (atom and bond types) we apply the discrete flow models (DFM) framework (Campbell et al., 2024), and for the continuous atom coordinates we apply the flow matching algorithm proposed by Tong et al. (2024), where a small amount of gaussian noise is added the interpolated coordinates x_t. Our model also predicts the formal charge for each atom but these do not participate in the generative flow matching process. In the remainder of this section we outline the full training and sampling procedure for SEMLAFLOW with molecular structures.

Training SEMLAFLOW As shown in existing conditional flow matching frameworks (Lipman et al., 2023; Albergo and Vanden-Eijnden, 2023; Campbell et al., 2024), training proceeds by firstly sampling: noise $z_0 \sim p_{\text{noise}}(z_0)$; data $z_1 \sim p_{\text{data}}(z_1)$; and a time $t \in [0, 1]$, and using these to sample from the time-dependent conditional flow $z_t \sim p_{t|1}(z|z_0, z_1)$. The joint molecular interpolation is therefore given as follows:

$$\mathbf{x}_t \sim \mathcal{N}(t\mathbf{x}_1 + (1-t)\mathbf{x_0}, \sigma^2) \qquad t \sim \text{Beta}(\alpha, \beta) \tag{14}$$

$$a_t \sim \text{Cat}(t\delta\{a_1, a_t\} + (1-t)\frac{1}{|\mathcal{A}|}) \tag{15}$$

$$b_t \sim \text{Cat}(t\delta\{b_1, b_t\} + (1-t)\frac{1}{|\mathcal{B}|}) \tag{16}$$

Where $\mathcal{A}$ and $\mathcal{B}$ are the sets of atom and bond types, respectively, and $\delta\{i, j\}$ is the Kronecker delta which is 1 when $i = j$ and 0 otherwise. We use $(\alpha, \beta) = (2.0, 1.0)$ and $\sigma = 0.2$ for all SEMLAFLOW models presented in this paper. In practice, we apply the equivariant optimal transport (OT) (Klein et al., 2024; Song et al., 2023) transformation to the sampled coordinates $\hat{\mathbf{x}}_0 = f_\pi(\mathbf{x}_0, \mathbf{x}_1)$ before sampling the interpolated value $\mathbf{x}_t$. This corresponds to applying a permutation and rotation which minimises the transport cost (in this case the mean-squared error) between $\mathbf{x}_0$ and $\mathbf{x}_1$.

Previous work on molecular structure generation has found it beneficial to train models to predict data directly rather than noise (Le et al., 2024) or a vector field (Stark et al., 2024a). SEMLAFLOW is therefore trained to learn a distribution $p_{1|t}^\theta(z_1|z_t)$ which predicts clean data from interpolated data using a SEMLA model with parameters θ. After sampling a predicted molecule $\tilde{z}_1 \sim p_{1|t}^\theta(z_1|z_t)$ where $\tilde{z}_1 = (\tilde{\mathbf{x}}_1, \tilde{\mathbf{a}}_1, \tilde{\mathbf{b}}_1, \tilde{\mathbf{c}}_1)$, the model is trained with a mean-squared error loss function ($\mathcal{L}_{\text{MSE}}$) for coordinates, and cross-entropy losses ($\mathcal{L}_{\text{CE}}$) for atom

Table 1: Molecular generation results on QM9. Models are grouped into those which infer bonds from coordinates (top) and those which generate bonds directly (bottom). Since some models only publish the proportion of molecules which are both unique and valid, results marked * are estimates for uniqueness.

Model	Atom Stab ↑	Mol Stab ↑	Valid ↑	Unique ↑	NFE
EDM	98.7	82.0	91.9	98.9*	1000
GCDM	98.7	85.7	94.8	98.4*	1000
GFMDiff	98.9	87.7	96.3	98.8*	500
EquiFM	98.9	88.3	94.7	98.7*	210
GeoLDM	98.9	89.4	93.8	98.8	1000
MUDiff	98.8	89.9	95.3	99.1	1000
GeoBFN	99.3	93.3	96.9	95.4	2000
FlowMol	99.7	96.2	97.3	–	100
MiDi	99.8	97.5	97.9	97.6	500
EQGAT-diff	$\mathbf{99.9}_{\pm 0.0}$	$98.7_{\pm 0.18}$	$99.0_{\pm 0.16}$	$\mathbf{100.0}_{\pm 0.0}$	500
SemlaFlow (Ours)	$\mathbf{99.9}_{\pm 0.0}$	$\mathbf{99.7}_{\pm 0.03}$	$\mathbf{99.4}_{\pm 0.03}$	$95.4_{\pm 0.12}$	100

types, bond types and charges. The final loss for the model is then given by the weighted sum:

$$\mathcal{L}_{\text{SemlaFlow}} = \lambda_x \mathcal{L}_{\text{MSE}}(\tilde{\mathbf{x}}_1, \mathbf{x}_1) + \lambda_a \mathcal{L}_{\text{CE}}(\tilde{\mathbf{a}}_1, \mathbf{a}_1) \quad (17)$$

$$+\lambda_b \mathcal{L}_{\text{CE}}(\tilde{\mathbf{b}}_1, \mathbf{b}_1) + \lambda_c \mathcal{L}_{\text{CE}}(\tilde{\mathbf{c}}_1, \mathbf{c}_1) \quad (18)$$

We also make use of self-conditioning, which was originally proposed for diffusion models as way of reusing the model's previous prediction when sampling (Chen et al., 2023). To create a self-conditioned SemlaFlow model, we adopt the same training procedure as HarmonicFlow (Stark et al., 2024a). We provide further details on this, along with the hyperparameters used for SemlaFlow, in Appendix C.

Sampling Molecules Once we have trained a SemlaFlow model, molecules can be generated by, firstly, sampling noise $z_0 \sim p_{\text{noise}}(z_0|n)$, and then integrating the ODE corresponding to the conditional flow $p_{t|1}$ from $t = 0$ to $t = 1$. For coordinates, the vector field corresponding to our choice of conditional flow is given by $\tilde{\mathbf{x}}_1 - \mathbf{x}_0 = \frac{1}{1-t}(\tilde{\mathbf{x}}_1 - \mathbf{x}_t)$ where $\tilde{\mathbf{x}}_1 \sim p^{\theta}_{1|t}$ as shown above. We then apply an Euler solver to integrate the ODE with step sizes Δt as follows: $\mathbf{x}_{t+\Delta t} = \mathbf{x}_t + \frac{\Delta t}{1-t}(\tilde{\mathbf{x}}_1 - \mathbf{x}_t)$. We refer readers to DFM (Campbell et al., 2024) for the sampling procedure for atom and bond types. In practice, we found that taking logarithmically spaced steps, where the model spends more time in parts of the vector field closer to $t = 1$, resulted in better performance than using constant step sizes.

5 EXPERIMENTS

In this section we provide results on benchmark 3D molecular generation tasks and compare the performance of our model to existing state-of-the-art approaches. In Appendix A we also provide results on multiple ablation experiments which demonstrate the impact of various contributions of this paper. One experiment demonstrates how controlling the latent message size d_l can lead to a significant model speedup without impacting performance. Another provides a side-by-side comparison of the Semla architecture with the EQGAT and EGNN architectures trained using identical flow matching training setups. We show that Semla outperforms the EQGAT and EGNN architectures at various model sizes, while also providing between 3 and 5 times faster inference over larger models. Finally, we provide samples from a SemlaFlow model trained on GEOM Drugs in Appendix E.

Evaluation Setup Two benchmark datasets, QM9 (Ramakrishnan et al., 2014) and GEOM Drugs (Axelrod and Gomez-Bombarelli, 2022), are used to assess SemlaFlow's abilities as an unconditional molecular generator. Since QM9 contains only very small molecules GEOM Drugs serves a more useful benchmark for distinguishing model performance. For both datasets we use the same data splits as MiDi and EQGAT-diff. To improve training times, however, we discard molecules with more than 72 atoms from the GEOM Drugs training set. This corresponds to about 1% of the training data; validation and test sets are left unchanged. All metrics for SemlaFlow presented below are calculated by sampling from the distribution of molecule sizes in the test set, and then generating molecules with the sampled number of atoms by integrating the trained ODE.

We compare SemlaFlow to a number of recently proposed models for 3D molecular generation. EDM (Hoogeboom et al., 2022), GCDM (Morehead and Cheng, 2024), GFMDiff (Xu et al., 2024), MUD-

Table 2: Molecular generation results on GEOM Drugs. Results for models which perform bond inference are provided in the appendix since they often do not provide results for all standard benchmark metrics.

Model	Atom Stab ↑	Mol Stab ↑	Valid ↑	Unique ↑	Novel ↑	NFE
FlowMol	99.0	67.5	51.2	–	–	100
MiDi	**99.8**	91.6	77.8	**100.0**	**100.0**	500
EQGAT-diff	**99.8**$_{\pm 0.0}$	93.4$_{\pm 0.21}$	**94.6**$_{\pm 0.24}$	**100.0**$_{\pm 0.0}$	99.9$_{\pm 0.07}$	500
SemlaFlow (Ours)	**99.8**$_{\pm 0.0}$	**97.3**$_{\pm 0.08}$	93.9$_{\pm 0.19}$	**100.0**$_{\pm 0.0}$	99.6$_{\pm 0.03}$	100

iff (Hua et al., 2024), GeoLDM (Xu et al., 2023) and GeoBFN (Song et al., 2024) are all diffusion-based models which infer bonds from atom positions. We also compare to EquiFM (Song et al., 2023) which uses flow-matching along with equivariant optimal transport to generate atom types and coordinates; bonds are then inferred based on these. FlowMol (Dunn and Koes, 2024) is a recently proposed flow matching model which learns a joint distribution over atoms and bonds and is probably the most similar existing work to ours. Finally, we also compare to MiDi (Vignac et al., 2023) and EQGAT-diff (Le et al., 2024) which we regard as the existing state-of-the-art. We use standard benchmark evaluation metrics: *atom stability*; *molecule stability*; *validity*; *uniqueness*; and *novelty*, which have been thoroughly described in previous works. We also provide a full description of these metrics in Appendix D. Results for models we evaluated are given as an average over 3 runs, sampling 10,000 molecules on each run, with standard deviations provided in subscripts. We also provide the number of function evaluations (NFE) required to sample one batch of molecules.

Molecular Generation Results Table 1 compares the performance of SemlaFlow with existing approaches on the QM9 dataset. Following Vignac and Frossard (2022); Hoogeboom et al. (2022) we do not provide novelty scores on QM9. SemlaFlow is trained for 300 epochs on QM9 on a single Nvidia A100 GPU. From the table we can see that SemlaFlow matches or exceeds all models on all metrics other than uniqueness, despite using 5 times fewer sampling steps than MiDi and EQGAT-diff. Our model also outperforms EquiFM and FlowMol, the only other flow matching models in the table.

Table 2 compares SemlaFlow's performance on GEOM Drugs to existing models which are also trained to generate molecular bonds. SemlaFlow matches or exceeds the performance of existing models on atom and molecule stability and produces only slightly fewer valid molecules than EQGAT-diff. The performance difference between FlowMol and SemlaFlow is also much more noticeable than on QM9; only two-thirds of molecules produced by FlowMol satisfy the molecular stability metric, compared to more than 97% for SemlaFlow. In addition to requiring significantly fewer evaluation steps than MiDi and EQGAT-diff, our model also requires much less compute for training. SemlaFlow trains for 200 epochs on a single Nvidia A100 GPU, compared to 800 epochs with 4 GPUs for EQGAT-diff.

Due to space constraints and since these models often do not provide results for all standard metrics, we provide results for models which infer bonds in Appendix B. Notably, while some of these models have higher validity than SemlaFlow, most have molecular stabilities close to 0 – they struggle to generate molecules with the correct number of bonds for each atom, especially on larger, drug-like molecules, like those in GEOM Drugs. Since the validity metric only measures whether a molecule can be loaded into RDKit (Gred Landrum *et al.*, 2023), we consider molecule stability a more comprehensive metric for this task. We discuss this issue further, along with other problems with existing evaluation metrics, in Appendix D.

Further Evaluation The validity and stability metrics presented here, however, only measure the topological structure of the molecule (i.e. the molecular graph); they provide no information on the quality of the generated conformations. We can also see that some existing models have already saturated these metrics on both QM9 and GEOM Drugs, warranting the introduction of additional evaluation methods. To further compare model performance and to allow evaluation of 3D generation, we introduce energy per atom and strain energy per atom as new benchmark metrics for this task. The energy measures the quality of a conformer, considering typical bonded and non-bonded interactions. The energy $U(\mathbf{x})$ of a conformation is inversely related to its probability according to the Boltzmann distribution $p(\mathbf{x}) = Z^{-1} \exp\left(-U(\mathbf{x})/kT\right)$ where T is the temperature and k is the Boltzmann constant. The strain is given by the difference $U(\mathbf{x}) - U(\tilde{\mathbf{x}})$ where $\tilde{\mathbf{x}}$ is the *relaxed* (i.e. minimised) conformation for $\mathbf{x}$. Since molecular energy is generally calculated as a sum of atomic energies we normalise both metrics by the number of atoms in each generated molecule. We argue that

Table 3: Comparison between EQGAT-diff and SemlaFlow with different numbers of sampling steps. Energy and strain energy are given as an average per atom and are measured in kcal $\cdot$ mol^{-1}. Sample time is measured by the average number of seconds to generate 1000 molecules. All metrics are averaged over 3 runs.

Model	Mol Stab ↑	Valid ↑	Energy ↓	Strain ↓	Sample Time ↓	NFE
EQGAT-diff	$93.4_{\pm 0.21}$	$94.6_{\pm 0.24}$	$3.38_{\pm 0.020}$	$3.23_{\pm 0.020}$	2293.0	500
SemlaFlow$_{20}$	$95.3_{\pm 0.14}$	$93.0_{\pm 0.10}$	$2.58_{\pm 0.018}$	$1.76_{\pm 0.007}$	20.3	20
SemlaFlow$_{50}$	$97.0_{\pm 0.21}$	$93.9_{\pm 0.12}$	$2.33_{\pm 0.044}$	$1.46_{\pm 0.043}$	49.8	50
SemlaFlow$_{100}$	$97.3_{\pm 0.08}$	$93.9_{\pm 0.19}$	$2.24_{\pm 0.028}$	$1.37_{\pm 0.022}$	99.3	100
Data	100.0	100.0	1.12	0.36	–	–

these metrics provide a very useful overview of the quality of the generated conformations and directly include measurements such as bond lengths and bond angles which have been proposed previously (Vignac et al., 2023; Buttenschoen et al., 2024). We use RDKit (Gred Landrum *et al.*, 2023) with an MMFF94 (Halgren, 1996) forcefield to calculate the energies and perform the minimisation.

In Table 3 we provide a performance and sampling time comparison between EQGAT-diff and SemlaFlow with varying numbers of ODE integration steps. SemlaFlow produces higher molecule stabilities and better energies and strain energies than EQGAT-diff, even with as few as 20 integration steps, although the validity of molecules produced by EQGAT-diff is slightly higher. Using SemlaFlow with 100 sampling steps corresponds to more than a 20x speedup over EQGAT-diff, and using 20 sampling steps corresponds to a two order-of-magnitude increase in sampling speed. Notably, however, molecules generated by EQGAT-diff have lower minimised energies than SemlaFlow, suggesting that their model is better at finding molecular conformations which have lower energy minima, while our model is better at producing lower strain energies.

6 RELATED WORK

3D Molecular Generation In addition to the unconditional molecular generators we outlined above, a number of works have attempted to directly generate ligands within protein pockets (Peng et al., 2022; Guan et al., 2023; Schneuing et al., 2022; Cremer et al., 2024; Ziv et al., 2024). However, these models also suffer from the issues we outlined previously, including long-sampling times (100s - 1000s of seconds for 100 molecules (Schneuing et al., 2022)) and generating invalid chemical structures or molecules with very high strain energies (Harris et al., 2023; Buttenschoen et al., 2024). GraphBP (Liu et al., 2022), an autoregressive model for protein-conditioned generation, is able to generate ligands faster, but suffers from poor docking scores in comparison to more recent diffusion models.

Flow Matching for 3D Structures Outside of small molecule design flow-matching has recently gained traction with generative models for biomolecules. FoldFlow (Bose et al., 2024) and FrameFlow (Yim et al., 2023) are both recently introduced flow matching models for protein structure generation. Multiflow (Campbell et al., 2024) attempts to jointly generate protein sequence and structure and introduces the discrete flow models (DFM) framework for flow matching generation of discrete data. Stark et al. (2024b) also introduce a framework for flow matching on discrete data, DirichletFM, and apply this to DNA sequence design. Finally, Verma et al. (2023) use a conjoined system of ODEs to train a model to jointly generate antibody sequences and structures.

7 CONCLUSION

In this work we have presented Semla, a novel equivariant message passing architecture exhibiting significantly better efficiency and scalability than existing approaches 3D generation. We have further introduced SemlaFlow, a flow matching model for jointly generating the topology and 3D conformations of molecular graphs. SemlaFlow achieves state-of-the-art results on 3D molecular generation benchmarks with two orders-of-magnitude faster sampling times. We have also highlighted issues with current molecular generation evaluation metrics and proposed the use of energy per atom and strain energy per atom for evaluating the quality of generated molecular conformations.

While we believe our model has made significant progress in solving key challenges for 3D molecular generators, many challenges remain. Firstly, the energies of the molecules generated by SemlaFlow are still significantly higher than that of the dataset; generating molecular coordinates with very high fidelity remains a problem for these models. Including further inductive biases or fine-tuning against an energy model could be an avenue to improve this in future work (Noé et al., 2019; Schreiner et al., 2023; Viguera Diez et al., 2024). Additionally, while SemlaFlow has shown significant

efficiency improvements over existing methods, it still uses a fully-connected message passing component, limiting it's scalability to larger molecular systems. We leave the further enhancement of the scalability of SEMLA to future work. We believe our model makes crucial step towards the practical application of 3D molecular generators, although we leave the integration of SEMLAFLOW into drug discovery workflows, either through RL-based fine-tuning or protein pocket conditioned generation, to future work.

Acknowledgements

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. Preliminary experiments were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS, project: 2024/22-33), partially funded by the Swedish Research Council through grant agreement no. 2022-06725. The authors thank T. Le (Pfizer) for sharing code and weights of Le et al. (2024) ahead of publication.

References

Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

Michael Samuel Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=li7qeBbCR1t.

Sara Romeo Atance, Juan Viguera Diez, Ola Engkvist, Simon Olsson, and Rocío Mercado. De novo drug design using reinforcement learning with graph-based deep generative models. *Journal of Chemical Information and Modeling*, 62(20):4863–4872, October 2022.

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

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Alberto Bietti, Luca Venturi, and Joan Bruna. On the sample complexity of learning under geometric stability. *Advances in neural information processing systems*, 34:18673–18684, 2021.

Thomas Blaschke, Josep Arús-Pous, Hongming Chen, Christian Margreitter, Christian Tyrchan, Ola Engkvist, Kostas Papadopoulos, and Atanas Patronov. Reinvent 2.0: an ai tool for de novo drug design. *Journal of chemical information and modeling*, 60 (12):5918–5922, 2020.

Joey Bose, Tara Akhound-Sadegh, Guillaume Huguet, Kilian FATRAS, Jarrid Rector-Brooks, Cheng-Hao Liu, Andrei Cristian Nica, Maksym Korablyov, Michael M. Bronstein, and Alexander Tong. SE(3)-stochastic flow matching for protein backbone generation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=kJFIH23hXb.

Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In *International Conference on Learning Representations*, 2021.

Martin Buttenschoen, Garrett M Morris, and Charlotte M Deane. Posebusters: Ai-based docking methods fail to generate physically valid poses or generalise to novel sequences. *Chemical Science*, 15(9): 3130–3139, 2024.

Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=kQwSbv0BR4.

Ting Chen, Ruixiang ZHANG, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. In *The Eleventh International Conference on Learning Representations*, 2023.

Julian Cremer, Tuan Le, Frank Noé, Djork-Arné Clevert, and Kristof T. Schütt. Pilot: equivariant diffusion for pocket-conditioned de novo ligand generation with multi-objective guidance via importance sampling. *Chem. Sci.*, 15:14954–14967, 2024. doi: 10.1039/D4SC03523B. URL http://dx.doi.org/10.1039/D4SC03523B.

Ian Dunn and David Ryan Koes. Mixed continuous and categorical flow matching for 3d de novo molecule generation. *ArXiv*, 2024.

Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, 107:3–11, 2018.

Victor Garcia Satorras, Emiel Hoogeboom, Fabian Fuchs, Ingmar Posner, and Max Welling. E (n) equivariant normalizing flows. *Advances in Neural Information Processing Systems*, 34:4181–4192, 2021.

Gred Landrum *et al.* Rdkit: Open-source cheminformatics. https://www.rdkit.org, 2023.

Jiaqi Guan, Wesley Wei Qian, Xingang Peng, Yufeng Su, Jian Peng, and Jianzhu Ma. 3d equivariant diffusion for target-aware molecule generation and affinity prediction. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=kJqXEPXMsE0.

Thomas A Halgren. Merck molecular force field. i. basis, form, scope, parameterization, and performance of mmff94. *Journal of computational chemistry*, 17(5-6): 490–519, 1996.

Charles Harris, Kieran Didi, Arian R Jamasb, Chaitanya K Joshi, Simon V Mathis, Pietro Lio, and Tom Blundell. Benchmarking generated poses: How rational is structure-based drug design with generative models? *arXiv preprint arXiv:2308.07413*, 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Emiel Hoogeboom, Vıctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *International Conference on Machine Learning*, pages 8867–8887. PMLR, 2022.

Chenqing Hua, Sitao Luan, Minkai Xu, Zhitao Ying, Jie Fu, Stefano Ermon, and Doina Precup. Mudiff: Unified diffusion for complete molecule generation. In *Learning on Graphs Conference*, pages 33–1. PMLR, 2024.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Leon Klein, Andreas Krämer, and Frank Noé. Equivariant flow matching. *Advances in Neural Information Processing Systems*, 36, 2024.

Jonas Köhler, Leon Klein, and Frank Noe. Equivariant flows: Exact likelihood generative learning for symmetric densities. In *International Conference on Machine Learning*, pages 5361–5370. PMLR, 2020.

Tuan Le, Frank Noe, and Djork-Arné Clevert. Representation learning on biomolecular structures using equivariant graph attention. In *Learning on Graphs Conference*, pages 30–1. PMLR, 2022.

Tuan Le, Julian Cremer, Frank Noe, Djork-Arné Clevert, and Kristof T Schütt. Navigating the design space of equivariant diffusion-based generative models for de novo 3d molecule generation. In *The Twelfth International Conference on Learning Representations*, 2024.

Yi-Lun Liao and Tess Smidt. Equiformer: Equivariant graph attention transformer for 3d atomistic graphs. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=KwmPfARgOTD.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Meng Liu, Youzhi Luo, Kanji Uchino, Koji Maruhashi, and Shuiwang Ji. Generating 3d molecules for target protein binding. In *International Conference on Machine Learning*, pages 13912–13924. PMLR, 2022.

Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

Hannes H Loeffler, Jiazhen He, Alessandro Tibo, Jon Paul Janet, Alexey Voronov, Lewis H Mervin, and Ola Engkvist. Reinvent 4: Modern ai–driven generative molecule design. *Journal of Cheminformatics*, 16(1):20, 2024.

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

Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457):eaaw1147, 2019.

Xingang Peng, Shitong Luo, Jiaqi Guan, Qi Xie, Jian Peng, and Jianzhu Ma. Pocket2mol: Efficient molecular sampling based on 3d protein pockets. In *International Conference on Machine Learning*, pages 17644–17655. PMLR, 2022.

Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):1–7, 2014.

Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=ryQu7f-RZ.

Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pages 9323–9332. PMLR, 2021.

Lisa Schneckenreiter, Richard Freinschlag, Florian Sestak, Johannes Brandstetter, Günter Klambauer, and Andreas Mayr. Gnn-vpa: A variance-preserving aggregation strategy for graph neural networks. In *The Second Tiny Papers Track at ICLR 2024*, 2024.

Arne Schneuing, Yuanqi Du, Charles Harris, Arian Jamasb, Ilia Igashov, Weitao Du, Tom Blundell, Pietro Lió, Carla Gomes, Max Welling, et al. Structure-based drug design with equivariant diffusion models. *arXiv preprint arXiv:2210.13695*, 2022.

Mathias Schreiner, Ole Winther, and Simon Olsson. Implicit transfer operator learning: Multiple time-resolution models for molecular dynamics. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=1kZx7JiuA2.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

Yuxuan Song, Jingjing Gong, Minkai Xu, Ziyao Cao, Yanyan Lan, Stefano Ermon, Hao Zhou, and Wei-Ying Ma. Equivariant flow matching with hybrid probability transport for 3d molecule generation. *Advances in Neural Information Processing Systems*, 36, 2023.

Yuxuan Song, Jingjing Gong, Hao Zhou, Mingyue Zheng, Jingjing Liu, and Wei-Ying Ma. Unified generative modeling of 3d molecules with bayesian flow networks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=NSVtmmzeRB.

Hannes Stark, Bowen Jing, Regina Barzilay, and Tommi Jaakkola. Harmonic self-conditioned flow matching for joint multi-ligand docking and binding site design. In *International Conference on Machine Learning*. PMLR, 2024a.

Hannes Stark, Bowen Jing, Chenyu Wang, Gabriele Corso, Bonnie Berger, Regina Barzilay, and Tommi Jaakkola. Dirichlet flow matching with applications to DNA sequence design. In *Forty-first International Conference on Machine Learning*, 2024b. URL https://openreview.net/forum?id=syXFAVqx85.

Behrooz Tahmasebi and Stefanie Jegelka. Sample complexity bounds for estimating probability divergences under invariances. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=sKjcrAC4eZ.

Alexander Tong, Kilian FATRAS, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=CD9Snc73AW. Expert Certification.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Yogesh Verma, Markus Heinonen, and Vikas Garg. Abode: Ab initio antibody design using conjoined odes. In *International Conference on Machine Learning*, pages 35037–35050. PMLR, 2023.

Clement Vignac and Pascal Frossard. Top-n: Equivariant set and graph generation without exchangeability. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=-Gk_IPJWvk.

Clement Vignac, Nagham Osman, Laura Toni, and Pascal Frossard. Midi: Mixed graph and 3d denoising diffusion for molecule generation. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 560–576. Springer, 2023.

Juan Viguera Diez, Sara Romeo Atance, Ola Engkvist, and Simon Olsson. Generation of conformational ensembles of small molecules via surrogate model-assisted molecular dynamics. *Machine Learning: Science and Technology*, 5(2):025010, April 2024. ISSN 2632-2153. doi: 10.1088/2632-2153/ad3b64. URL http://dx.doi.org/10.1088/2632-2153/ad3b64.

Can Xu, Haosen Wang, Weigang Wang, Pengfei Zheng, and Hongyang Chen. Geometric-facilitated denoising diffusion model for 3d molecule generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 338–346, 2024.

Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric diffusion model for molecular conformation generation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=PzcvxEMzvQC.

Minkai Xu, Alexander S Powers, Ron O Dror, Stefano Ermon, and Jure Leskovec. Geometric latent diffusion models for 3d molecule generation. In *International Conference on Machine Learning*, pages 38592–38610. PMLR, 2023.

Jason Yim, Andrew Campbell, Andrew YK Foong, Michael Gastegger, José Jiménez-Luna, Sarah Lewis, Victor Garcia Satorras, Bastiaan S Veeling, Regina Barzilay, Tommi Jaakkola, et al. Fast protein backbone generation with se (3) flow matching. *arXiv preprint arXiv:2310.05297*, 2023.

Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip H.S. Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16259–16268, October 2021.

Yael Ziv, Brian Marsden, and Charlotte M Deane. Molsnapper: Conditioning diffusion for structure based drug design. *bioRxiv*, pages 2024–03, 2024.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Not Applicable]
   (b) Complete proofs of all theoretical results. [Not Applicable]
   (c) Clear explanations of any assumptions. [Not Applicable]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Yes]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A ABLATION EXPERIMENTS

This section includes additional experiments which were designed to explain the design decisions made in the SEMLA architecture and to assess the benefit of using SEMLA over other equivariant architectures for molecular generation.

A.1 Latent Message Size

Firstly, Table 4 assesses the impact of the d_l hyperparameter which specifies the size of the latent dimension when performing latent attention. Each model was trained on GEOM Drugs for 100 epochs using the same training setup as the SEMLAFLOW model in Table 2.

While $d_l = 256$ produces the best overall performance in terms of molecule stability, validity and energy, the model takes significantly longer to sample the same number of molecules. Crucially, the difference in performance between all four models is relatively small, despite the notable differences in sample time. For this reason we selected $d_l = 64$ for all SEMLAFLOW models.

Table 4: Comparison of SEMLAFLOW models with different sizes of latent attention dimension d_l. All models are trained in identical conditions for 100 epochs. Sample time is measured by the average number of seconds to generate 1000 molecules. All metrics are averaged over 3 runs.

d_l	Mol Stab ↑	Valid ↑	Energy ↓	Strain ↓	Sample Time ↓
32	$97.7_{\pm 0.13}$	$92.7_{\pm 0.30}$	$2.73_{\pm 0.014}$	$1.80_{\pm 0.004}$	89.0
64	$97.6_{\pm 0.02}$	$93.1_{\pm 0.30}$	$2.63_{\pm 0.019}$	$1.70_{\pm 0.013}$	98.3
128	$97.1_{\pm 0.05}$	$92.8_{\pm 0.07}$	$2.87_{\pm 0.026}$	$1.98_{\pm 0.022}$	122.0
256	$98.4_{\pm 0.06}$	$94.6_{\pm 0.14}$	$2.61_{\pm 0.006}$	$1.71_{\pm 0.008}$	217.7

A.2 Architecture Ablation

We also wish to compare the generative performance and sampling efficiency of our proposed SEMLA architecture with existing E(3)-equivariant architectures for 3D molecular generation. To do this we train various network architectures within a consistent experimental setup – we apply the same flow matching training and inference procedure as SEMLAFLOW but swap out SEMLA for existing equivariant architectures. We benchmark SEMLA against the 4 layer and 9 layer versions of EGNN (Satorras et al., 2021) proposed in EDM (Hoogeboom et al., 2022), as well as the EQGAT network (Le et al., 2022) used in the state-of-the-art EQGAT-diff model Le et al. (2024). We also provide a comparison with an expanded, 16 layer version of EGNN, which has roughly the same number of parameters as SEMLA. In order to handle bond types, we modify the pairwise MLPs for EGNN on the first and last layer of the network in the same way as SEMLA. Each architecture is given a fixed training budget of 24 hours. Since existing models are not setup for self-conditioned inputs, all models, including SEMLA, are trained without self conditioning.

Table 5: Comparison of different architectures all trained on an identical flow matching setup. Sample time is measured by the average number of seconds to generate 1000 molecules. All metrics are averaged over 3 runs.

Architecture	Mol Stab ↑	Valid ↑	Energy ↓	Strain ↓	Sample Time ↓
EGNN (4 layer)	$71.4_{\pm 0.76}$	$55.1_{\pm 0.06}$	$10.89_{\pm 0.059}$	$10.03_{\pm 0.063}$	69.3
EGNN (9 layer)	$94.0_{\pm 0.16}$	$87.9_{\pm 0.40}$	$4.04_{\pm 0.047}$	$3.13_{\pm 0.023}$	151.4
EGNN (16 layer)	$94.7_{\pm 0.06}$	$89.9_{\pm 0.06}$	$3.97_{\pm 0.079}$	$3.07_{\pm 0.065}$	532.0
EQGAT	$97.1_{\pm 0.07}$	$83.9_{\pm 0.33}$	$4.19_{\pm 0.013}$	$3.32_{\pm 0.019}$	337.4
SEMLA (Ours)	$96.3_{\pm 0.14}$	$91.2_{\pm 0.26}$	$3.04_{\pm 0.014}$	$2.15_{\pm 0.014}$	99.4

The results of this architecture ablation study are shown in Table 5. SEMLA shows better validities and significantly better energies than existing architectures, although has slightly lower molecule stability than EQGAT. Crucially though SEMLA is more than 3 times faster than EQGAT and more than 5 times faster than the similarly-sized 16 layer EGNN network while still producing molecules of comparable or higher quality.

B ADDITIONAL RESULTS ON GEOM DRUGS

In this section we include additional results of existing models on GEOM Drugs which we were not able to fit into the main text. We also include the results from Table 2 for full comparison. Many models have been proposed for this task recently and we chose to focus our comparison in the main text to models which learn to generate bonds rather than inferring them from coordinates.

Table 6 shows that while some models which infer bonds (shown in the top segment of the table) produce higher validities than those which do not, their atom and molecule stabilities are often significantly lower. However, since validity only measures whether a molecule can be sanitised by RDKit, we consider molecule stability to be a much better measure of the quality of the generated molecules. We include a further discussion and explanation for this in Appendix D.

Table 6: Additional results on GEOM Drugs, including models which infer bonds based on the generated coordinates. Since many of these models do not publish molecule stability results, numbers marked with * are estimates computed by taking AS^{44} where AS is atom stability and 44 is the average number of atoms in GEOM Drugs molecules. This follows the same procedure used in the EquiFM paper to estimate molecule stability.

Model	Atom Stab ↑	Mol Stab ↑	Valid ↑	Unique ↑	Novel ↑	NFE
EDM	81.3	0.0*	–	–	–	1000
GCDM	89.0	5.2	–	–	–	1000
MUDiff	84.0	60.9	98.9	–	–	1000
GFMDiff	86.5	3.9	–	–	–	500
EquiFM	84.1	0.0*	98.9	–	–	–
GeoBFN	86.2	0.0*	91.7	–	–	2000
GeoLDM	98.9	61.5*	**99.3**	–	–	1000
FlowMol	99.0	67.5	51.2	–	–	100
MiDi	**99.8**	91.6	77.8	**100.0**	**100.0**	500
EQGAT-diff	**99.8**$_{\pm 0.0}$	93.4$_{\pm 0.21}$	94.6$_{\pm 0.24}$	**100.0**$_{\pm 0.0}$	99.9$_{\pm 0.07}$	500
SemlaFlow (Ours)	**99.8**$_{\pm 0.0}$	**97.3**$_{\pm 0.08}$	93.9$_{\pm 0.19}$	**100.0**$_{\pm 0.0}$	99.6$_{\pm 0.03}$	100

C FURTHER MODEL AND TRAINING DETAILS

This section provides further detail on the design of the full Semla architecture, as well further training details including the hyperparameters used to trian QM9 and GEOM Drugs models.

C.1 The Semla Architecture

As mentioned in Section 3, we do not carry pairwise edge features through the model, but rather encode the edge information into the node features on the first layer and then generate edge features on the final layer. After these initial edge features are generated they are passed through a learnable refinement component where the final predicted coordinates are also given as input, along with the final invariant node features. The remaining components of the model are used to encode atom and bond types and predict distributions for atom types, bond types and formal charges. A full overview of a Semla model is shown in Figure 3.

C.2 Training Details

All models were trained with the AMSGrad (Reddi et al., 2018) variant of the Adam optimiser (Kingma and Ba, 2014) with a learning rate (LR) of 0.0003. We also apply linear LR warm-up, using 2000 warm-up steps for QM9 and 10000 warm-up steps for GEOM Drugs. During training we clip the norms of the gradients at 1.0 for all models. Loss weightings $(\lambda_x, \lambda_a, \lambda_b, \lambda_c) = (1.0, 0.2, 0.5, 1.0)$ were used to train QM9 models. The same weightings were used for GEOM Drugs, except we set $\lambda_b = 1.0$.

When training with self-conditioning half of the training batches are treated as normal and the other half are trained on as self-conditioning batches. In this case the batch is firstly processed by the model to generate

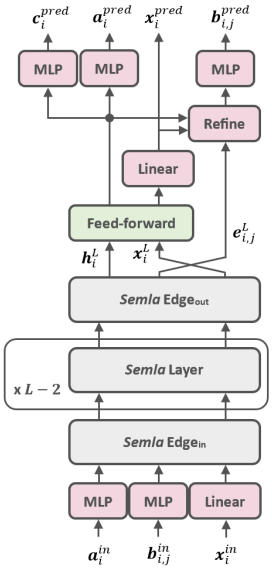

Figure 3: An overview of a full SEMLA model. A SEMLA model is created from a stack of SEMLA layers where the first layer encodes bond information into the node features and the final layer generates bond information from the pairwise features.

conditioning inputs, these are then detached from the computation graph, and finally used as conditioning inputs for the model training step. In practice the conditioning inputs are concatenated with the interpolated data and embedded at the start of the model. For atom and bond types the conditioning inputs are softmax-normalised probability distributions over the predicted categorical types.

In order to make the training as efficient as possible we place molecules in the dataset into buckets based on their size, and then form minibatches for training and evaluation within the buckets. This ensures that all batches contain similarly sized molecules so that the amount of padding within each batch is minimised. With this setup we can also apply a cost function to select the batch size for each bucket separately; since the amount of memory required to process a molecule increases quadratically with the number of atoms, this helps to balance the GPU memory consumption for each batch. In practice, though, we simply apply a linear cost function and use a batch size of 4096 atoms per batch for all SEMLAFLOW models. While we have found our bucketing scheme leads to a significant increase in training speed, it may also introduce additional bias into the training since molecules within each batch are no longer selected completely at random. Although we have not attempted to quantify this bias our results seem to show that bucketing is not significantly detrimental to performance.

D EVALUATION METRICS

In this section we outline a number of shortcomings with current evaluation for metrics for unconditional 3D generative models. Firstly, we provide a full description for each existing benchmark metric we have used:

- **Atom stability** measures the proportion of atoms which have the correct number of bonds, according to a pre-defined lookup table.
- **Molecule stability** then measures the proportion of generated molecules for which all atoms are stable.
- **Validity** is given by the proportion of molecules which can be successfully sanitised using RDKit.
- **Uniqueness** measures the proportion of generated molecules which are unique. This is calculated by comparing molecules based on their canonical SMILES representation.
- **Novelty** measures the proportion of generated molecules which are not in the training set.

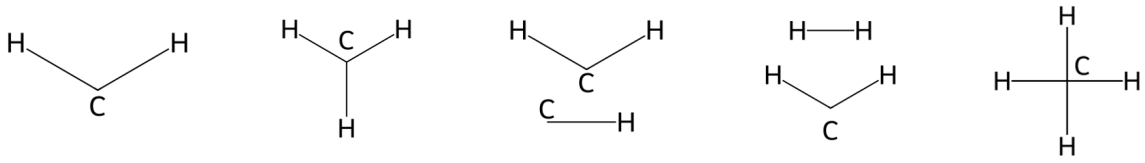

Figure 4: Examples of molecules which would be considered valid by RDKit.

D.1 Issues with Existing Metrics

While a lot of progress has been made recently in improving benchmarks for structure-based drug design (Harris et al., 2023; Buttenschoen et al., 2024), we believe many evaluation metrics used for unconditional 3D generation are still not fit for purpose. We hope that the introduction of energy per atom and strain energy per atom will help to alleviate this but believe it is still important to discuss the shortcomings of existing evaluation methods. In this section we highlight issues with the standard benchmark metrics for this task.

Validity Since RDKit is free to add implicit hydrogens and, in some cases, to modify the formal charge on atoms, RDKit validity tells us very little about the "correctness" of a molecule. This is shown explicitly in Figure 4; all molecules shown in the figure would be marked as valid by RDKit, even though only the rightmost molecule has the correct number of bonds. Additionally, the validity metric on its own is not capable of flagging disconnected molecules – generated molecules which contain multiple fragments which are not connected by any bond. The validity metric is, however, capable of spotting when a molecule has too many bonds.

Atom and Molecule Stability Different models sometimes use different lookup tables to compute atom (and therefore molecule) stability, potentially leading to different evaluation results for the same generated molecules. Often this depends on whether models predict formal charges for each atom or not, since those that do need to take the charge into account in the look-up table. Additionally, we have found that existing lookup tables used to define atom stability have little basis in chemical validity. As an example, the existing definition for atom stability allows an uncharged carbon atom with 3 (single covalent bond equivalent) bonds to be considered 'stable', although this has little-to-no chemically valid basis.

Conclusion We believe the issues highlighted here reflect the importance of using metrics such as energy which is able to quantify the quality of both the generated topology as well as the generated molecular conformation. We also hope that highlighting these issues will lead to more in-depth evaluations of 3D unconditional molecular generation models in the future and lay a foundation for more reproducible and coherent benchmarking.

E SAMPLES FROM SEMLAFLOW

In this section we present samples from SemlaFlow trained on GEOM Drugs. The samples were generated randomly but we have rotated them where necessary to aid visualisation.

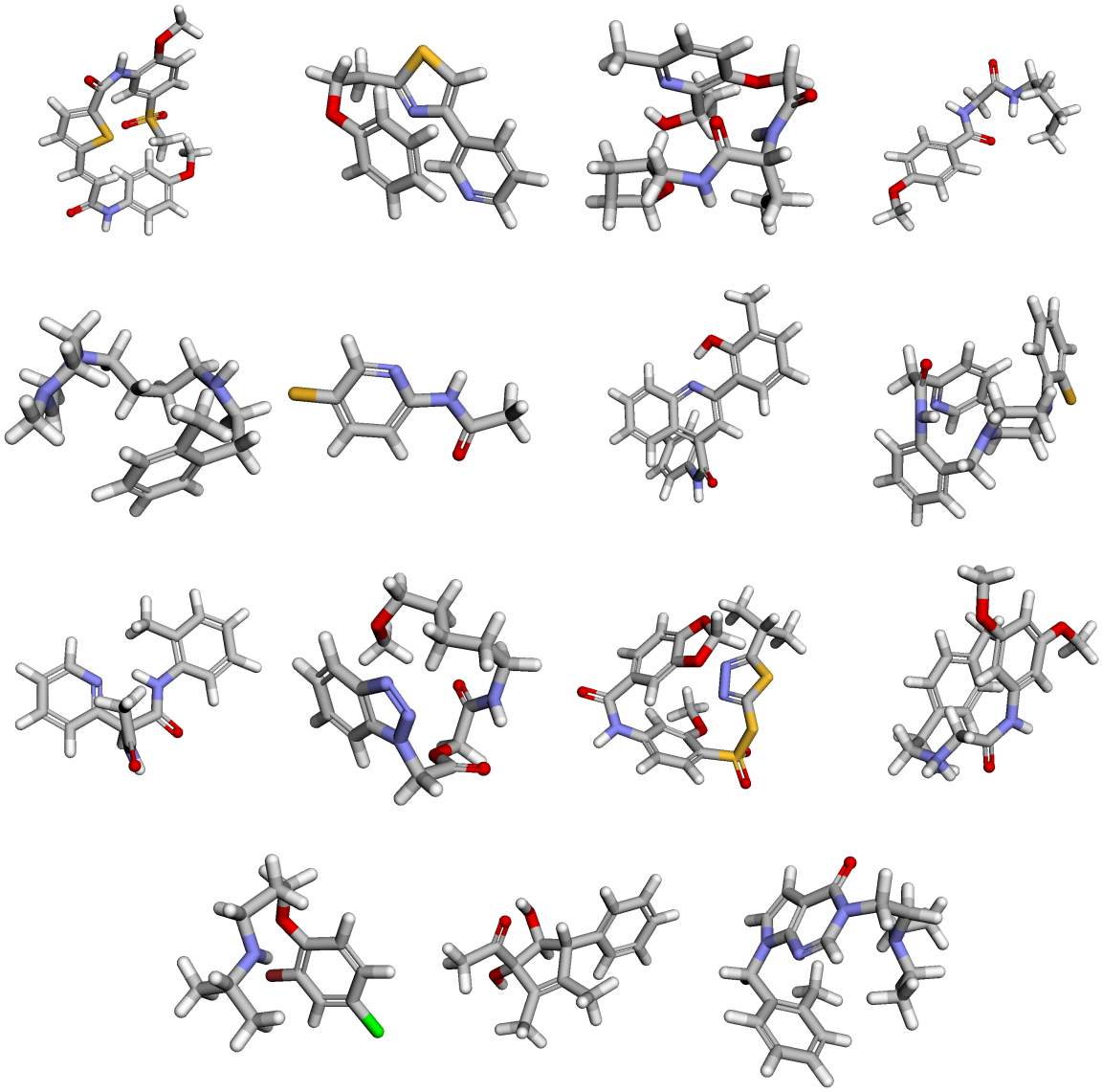

Figure 5: Random samples from a SemlaFlow model trained on GEOM Drugs. These samples were generated using 100 ODE integration steps.