Autoregressive Models Enable Efficient Conditional 3D Molecular Generation

Anonymous Authors[1]

Abstract

The reigning paradigm for small molecule 3D structure generation in recent years has been the so-called stochastic interpolant models, which includes the class of diffusion and flow-based generative models. These models learn how to transport samples from an easy-to-sample base distribution (such as a Gaussian distribution defined over $\mathbb{R}^{3N}$, where N is the number of atoms) to the distribution of 3D molecular structures. Critically, the number of atoms N needs to be sampled apriori before the learned transport process, as all atoms are transported simultaneously. This makes such models hard to use in tasks such as fragment completion, where generation must proceed from an incomplete molecule while the remaining number of atoms are unknown. Indeed, most benchmarks for small molecule 3D structure generation simply test unconditional generation, where the goal is simply to sample possible 3D molecular structures without any constraints. Unfortunately, existing metrics overly emphasize exact bond length and angular distribution matching. We argue that the key goal of molecule generation is *conditional* generation, where we wish to generate molecules conditional on some geometric or chemical constraints. We show that a long-forgotten approach of building molecules autoregressively actually performs favorably in these regimes. In fact, by carefully engineering the simple training recipe proposed in Symphony, we find that autoregressive molecule generative models can learn 1) significantly more efficiently than existing diffusion/flow-based models, 2) enable significantly more accurate conditional generation in terms of quantum mechanical properties, 3) enable simple and efficient fragment completion with high success rates.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Submitted to the 2026 Workshop on Generative and Agentic AI for Biology (ICML 2026). Do not distribute.

1. Introduction

Generative models have emerged as a promising method for navigating the vast landscape of chemical space (Anstine & Isayev, 2023). For example, generative models for 3D molecular structures enable sampling the 3D structure of a molecule, conditional on certain desired molecular properties. For example, AlphaFold (Jumper et al., 2021; Abramson et al., 2024) samples the crystallized 3D structure of a protein conditional on its amino acid residue sequence. We are particularly interested in the design of small organic molecules, which make up approximately 90% of all therapeutic drugs sold today (Makurvet, 2021). In this context, the atom identities are unknown, and the atoms themselves cannot be consistently ordered into a sequence. These constraints necessitate a different approach from those invoked by protein structure prediction models.

Formally, we wish to sample from the distribution p of all-atom 3D positions and atomic numbers, defined over $\mathbb{R}^{3N} \times \mathbb{Z}^N$, where N represents the number of atoms in the molecule, and is sampled over some distribution p_N.

1.1. Stochastic Interpolant Models

The most prevalent paradigm for small molecule 3D structure generation in recent years have been the class of *stochastic interpolant* (Albergo et al., 2023) models, which generalize the class of flow-based (Lipman et al., 2023) and diffusion-based (Song et al., 2021) models. The key idea is to learn how to transport samples from an easy-to-sample base distribution p_0 to the target distribution p_1 of choice.

We illustrate the case where p_0 and p_1 are defined on Euclidean space $\mathbb{R}^d$ for some dimension d. Consider a sample $x_0 \sim p_0$ and a sample $x_1 \sim p_1$. We can define a time-dependent coupling: $x_t \equiv I(t, x_0, x_1) + \gamma(t)z$ where I is a (deterministic) interpolant function and $z \sim \mathcal{N}(0, \mathbb{I}_d)$ is Gaussian noise. I interpolates between x_0 and x_1 by satisfying the boundary conditions $I(0, x_0, x_1) = x_0, I(1, x_0, x_1) = x_1$. γ also satisfies the boundary conditions $\gamma(0) = \gamma(1) = 0$ as well as non-negativity $\gamma(t) \geq 0$. Then, x_t itself has a probability distribution which interpolates between p_0 and p_1, which we term p_t. Given any choice of stochastic interpolant, let us define the *velocity*

field b and the *score field* s as:

$$b(t,x) \equiv \mathbb{E}[x_1|x_t = x] \tag{1}$$

$$s(t,x) \equiv \nabla_x \log p_t(x) = -\frac{\mathbb{E}[z|x(t)=x]}{\gamma(t)} \tag{2}$$

Then, starting from a sample $x_0 \sim p_0$ and propagating the (stochastic) differential equation forward in time:

$$dx_t^{\mathrm{F}} = (b(t,x_t^{\mathrm{F}})dt + \epsilon(t)s(t,x_t^{\mathrm{F}}))dt + \sqrt{2\epsilon(t)}\,dW(t) \tag{3}$$

where W is the standard Wiener process and $\epsilon(t) \geq 0$ is arbitrary, from the results of Albergo et al. (2023), we obtain samples x_t^{F} which are distributed according to p_t at each t. In particular, we can obtain the required samples $x_1^{\mathrm{F}} \sim p_1$ by propagating upto $t = 1$. In practice, the true velocity field b and score field s cannot be obtained exactly; hence, they are approximated via neural networks $\hat{b}$ and $\hat{s}$, which can be learned via gradient descent by sampling $x_0 \sim p_0, x_1 \sim p_1, z \sim \mathcal{N}(0, \mathbb{I}_d)$ to approximate the expectations in Equation 1 and Equation 2. The learned velocity and score fields thus form the transport.

However, applying this modeling technique for 3D molecule generation comes with several challenges. In particular, while the 3D atomic positions lie in Euclidean space and are hence naturally handled by this framework (eg. by setting $d = 3N$), the discrete nature of the atomic numbers (or atom types) makes them trickier to model. One of the first approaches (Hoogeboom et al., 2022) was to simply embed the atom types as a one-hot encoding up to some maximum atomic number, artificially making the discrete atom types continuous. A more flexible approach is latent-space modelling as developed by Joshi et al. (2025); Geffner et al. (2025), where the atomic positions and atom types are encoded continuously in some higher dimension d' (usually via a variational autoencoder (Kingma & Welling, 2019)) and the stochastic interpolation happens in this latent space. Discrete stochastic interpolants (Potaptchik et al., 2026; Stark et al., 2024; Holderrieth et al., 2025) which naturally enable transport over discrete spaces via a continuous-time Markov chain with discrete states have also emerged as a powerful alternative. These methods have proven successful across a large variety of generative tasks and modalities. However, a key point that often remains unresolved is the necessity to sample and fix the number of atoms N before any transport can proceed. Indeed, the dimensionality d of the space is fixed during the transport. The most common approach is to simply estimate the distribution p_N of N from the training data. However, this is fundamentally limiting in the context of conditional generation, where the number of atoms N may also depend strongly on the type of conditioning applied. FlowMol3 (Dunn & Koes, 2025) and Schneuing et al. (2025) add fake atoms (with a new 'fake atom' type) anchored near real atoms at training time. This increases the robustness of the model but does not fully solve the core issue as the distribution of fake atoms cannot truly be well-defined. Havasi et al. (2025) recently made some progress in the language modelling context by allowing the model to mask out and delete tokens, however, the efficacy of those techniques has not been demonstrated yet in the context of molecular structure modelling. Recently, Billera et al. (2026) introduced Branching Flows, where the elements learn branching and deletion rates, enabling arbitrary-length generation with preliminary results on the small-molecule QM9 (Ramakrishnan et al., 2014) dataset and antibody sequence design. While extremely promising, their sampling scheme is signficantly more involved and will likely need further tricks to be competitive with state-of-the-art stochastic interpolant models.

The first models – G-SchNet (Gebauer et al., 2019), G-SphereNet (Luo & Ji, 2022) – for 3D molecule generation were indeed autoregressive, building molecules atom-by-atom. Symphony (Daigavane et al., 2024) leveraged higher-order $E(3)$-equivariant networks with a spherical harmonic projection mechanism to predict the next atom's position, similar in spirit to Simm et al. (2020; 2021). Quetzal (Cheng et al., 2025) improved the scalability of these autoregressive methods, demonstrating their applicability to variable-length tasks such as hydrogen decoration and scaffold completion. We build upon their benchmarks in our work here.

2. Datasets

We use the well-established QM9 (Ramakrishnan et al., 2014) dataset for development, which consists of 134000 molecules with atom types `H`, `C`, `N`, `O`, `F` and upto 9 heavy atoms. QM9 has rich quantum mechanical properties such as the polarizability α, heat capacity C_v and HOMO-LUMO gap Δ. We also report preliminary results on the GEOM-DRUGS (Axelrod & Gómez-Bombarelli, 2022) dataset.

We use the recently introduced `atomic-datasets` repository for standardized dataset processing and split definition. However, many of the baselines we compare touse different training splits and processing strategies. For example, SemlaFlow (Irwin et al., 2025) discards molecules with more than 72 atoms from the GEOM-DRUGS training set to improve their training time. This makes a fair comparison rather complicated; however, we still expect to see broad trends in model performance.

3. Methods

We start with the simple recipe from Symphony (Daigavane et al., 2024). The key idea is to iteratively build a molecule atom-by-atom. At any intermediate step in the generation

trajectory, the model predicts a focus atom among all existing atoms in the current molecular fragment, predicts the next species to be placed, and then predicts a probability distribution of the next atom position relative to the focus atom. The underlying architecture is an $E(3)$-equivariant message-passing geometric graph neural network (Daigavane et al., 2021) similar to NequIP (Batzner et al., 2022). This means that the model's predictions are invariant to translations but transform naturally under rotations.

However, we have made several improvements to the original Symphony recipe, aimed to improve performance and training stability:

- **Separating out the radial and angular components**: The original Symphony model predicted the 3D distribution $p(r, \theta, \phi)$ representing the target position relative to the focus atom in spherical coordinates r, θ, ϕ, which was essentially discretized over $\approx 600,000$ points in 3D space. This leads to significant GPU memory usage and slows down sampling. Instead, we factorize the 3D distribution as a radial distribution $p(r)$ and an angular distribution conditional on the radius $p(\theta, \phi \mid r)$. This significantly reduces GPU memory usage and enables scaling to much larger molecules than Symphony was capable of training on.
- **Separable convolutions**: We utilize separable (also called depth-wise) convolutions from EquiformerV2 (Liao et al., 2024) as implemented in the `e3tools` library (Kleinhenz & Daigavane, 2025). The idea is that each channel of the input node features is independently interacted with the edge spherical harmonics to create the output mode features, as opposed to the full convolution where all channels interact with each other. We add a separate linear layer to mix between the channels after the convolution. The separable convolution significantly improves both the throughput and reduces the peak GPU memory usage of the model.
- **Adding neighborhood information to the angular predictor**: We found that the model occasionally places atoms on top of each other, missing the fact that some atoms have already been placed. We solve this issue by adding the spherical harmonic projection of the Dirac delta function at neighborhood atoms of the focus node to the angular predictor, enabling it to avoid pre-existing atoms at negligible cost.
- **OpenEquivariance Kernels**: The FlashAttention (Dao et al., 2022) series of kernels for the attention module has significantly sped up both training and inference of the popular (non-equivariant) Transformer (Vaswani et al., 2017) model. A major bottleneck for equivariant networks has been the lack of dedicated engineering efforts to develop efficient CUDA-based implementations of equivariant primitives such as the Clebsch-Gordan tensor product. This has changed recently with the development of a Clebsch-Gordan tensor product kernel by the OpenEquivariance team; enabling unprecedented scaling of this class of $E(3)$-equivariant models beyond previous attempts (Geiger et al., 2024). These kernels are also supported in the `e3tools` library, and we simply need to replace `e3tools.SeparableConv` with `e3tools.FusedSeparableConv` for the appropriate layer.
- **Fragmentation strategies**: The original Symphony recipe decomposed molecules into fragments by choosing the first atom at random, then picking the nearest neighbor among all remaining atoms repeatedly, until the molecule is complete. We add some randomness to the nearest neighbor selection by selecting any atom within a small 0.5Å tolerance of the nearest neighbor distance helps the model's robustness. Further, we add several other fragmentation strategies: removing a random subset of atoms within the molecule, removing a random sphere of atoms within the molecule, dehydrogenation where all of the hydrogens are removed from the molecule. Each of the alternative strategies is chosen with probability 0.1 for a given molecule. We find that these fragmentation strategies significantly improve the robustness of the model.
- **Conditioning**: We add the ability to condition on (any dimensional) quantum mechanical properties by embedding the given conditioning, and then using them to predict scaling factors for the intermediate node features at each message-passing step.

Collectively, our improvements enable us to scale up to massive batch sizes of over 200000 edges on a single NVIDIA RTX A5500 GPU, enabling convergence on QM9 and GEOM-DRUGS within 50 GPU hours. This is an *order-of-magnitude less compute* than the 1200/500 GPU-hours needed by the state-of-the-art ADiT (Joshi et al., 2025) / Zatom-1(Morehead et al., 2026) models with *two orders-of-magnitude fewer parameters* (2M for Symphony++ versus 32M-450M/80M-300M for ADiT/Zatom-1) on more memory-restricted hardware (24 GB of DRAM on a NVIDIA RTX A5500 versus 32/80 GB of DRAM on a NVIDIA V100/A100 used by ADiT/Zatom-1 respectively.

Finally, our inference speeds are significantly faster than state-of-the-art models, requiring only ≈ 4 minutes to sample 10000 QM9-like molecules on a single NVIDIA RTX A5500 GPU, which is similar to the numbers reported by Zatom-1 on an NVIDIA A100 GPU. For comparison, ADiT takes ≈ 30 minutes while GCDM and GeoLDM take several hours in the same setting. We note that the sampling

times across models can vary significantly across various batch sizes and GPU memory bandwidths, making an exact comparison difficult.

Symphony explicitly models the probability distribution of the focus atom, target species, target radial distance, and target angular position, by predicting the unscaled logits of these distributions. These logits f are scaled by an inverse temperature β before being passed through a softmax operation to obtain a probability distribution p over the appropriate domain X:

$$p(x;\beta) = \frac{\exp(\beta f(x))}{\int_X \exp(\beta f(x'))dx'} \tag{4}$$

β inherently controls the randomness of the sampling process; a higher β leads to increased sampling from the modes, while a lower β encourages more exploratory sampling. This tradeoff is clearly captured in Figure 1 and tabulated in Table 5, where increasing β trades off validity for uniqueness.

4. Results

First, we show that our architectural and training improvements enable an efficient *unconditional* model.

4.1. Benchmarking Unconditional Generation on QM9

Table 1. Validity % per starting element and uniqueness % (measured using SMILES) for different inverse temperature β configurations. Each element serves as the starting atom for 200 generated molecules.

Inverse Temperature			Per-Element Validity (%)					Uniqueness (%)	
Focus	Radial	Angular	H	C	N	O	F	xyz2mol	SMILES
1.0	1.0	10.0	95.5	94.0	91.5	91.5	97.5	97.9	98.2
1.0	1.0	20.0	95.0	**95.0**	91.5	92.0	96.5	**99.0**	**99.0**
1.0	2.0	20.0	95.0	94.5	**94.5**	**93.5**	97.0	96.9	97.2
2.0	2.0	20.0	**99.5**	**97.0**	92.5	91.5	**98.5**	83.9	85.0

We start the sampling process from a single `C` atom. (In theory, we could also learn this from the data by predicting an initial starting species and placing it anywhere, since our model is translation-invariant.) Table 1 shows that the validity is high no matter which starting species is chosen in QM9. The validity is particularly high for the monovalent species `H` and `F`, which makes intuitive sense because the initial steps of the sampling process are easier to define.

Table 3 shows that Symphony++ significantly improves over existing autoregressive models – G-SphereNet, G-SchNet and its predecessor Symphony – with performance slightly below the state-of-the-art SemlaFlow, Zatom-1 and ADiT. Note that Symphony++ does not predict bonds like SemlaFlow, and generates all atoms including hydrogens. We obtain samples from each of these models and pass them to the `molmetrics` evaluation suite which provides an easy interface to the RDKit validity checker with `rdkit==2025.3.5` and PoseBusters (Buttenschoen et al., 2023) evaluation suite. Validity is measured in one of two ways; Daigavane et al. (2024) used `xyz2mol` (Kim & Kim, 2015) to assign bond orders and check that all atom valencies are correct, while Joshi et al. (2025); Morehead et al. (2026) simply check if RDKit can infer a valid SMILES string from the molecule. The latter is more popular in the literature (likely because it is a weaker metric), but we report both numbers for clarity.

Table 2. Effect of quick post-hoc geometry relaxation with Nequix-MP-1 (50 steps L-BFGS, $f_{\max} = 0.05$ eV/Å) on 500 Symphony++ generated molecules.

	Validity (%)		Stability (%)	
	`xyz2mol`	SMILES	Atom	Molecule
Symphony++	88.8	94.0	92.7	47.0
Symphony++ (relaxed)	**92.0**	**99.2**	**97.6**	**81.6**

As detailed by Nikitin et al. (2025), the atom and molecule stability metrics are quite strict and do not appropriately handle aromatic compounds. Although the validity and uniqueness metrics are correlated with overall molecule quality, at the top end, these metrics are essentially saturated. In particular, Table 2 shows that these metrics for our reasonable Symphony model can be improved with a quick relaxation with the pretrained Nequix-MP-1 (Koker et al., 2025) machine-learned interatomic potential to adjust bond lengths and angles. The key advantage of a machine-learned interatomic potential here is that they act on atom coordinates only, without requiring any bond resolution.

The Symphony++ molecules that were deemed valid score very highly on the PoseBusters (Buttenschoen et al., 2023) evaluation suite; in fact, Symphony++ scores the highest on the Internal Energy benchmark across all models.

4.2. Benchmarking Fragment-Conditional Generation on QM9

To further evaluate our model beyond these narrow metrics, we develop a *fragment completion* benchmark. The idea is to see if Symphony++ can fill in partially completed molecules correctly, essentially mimicking fragment-conditional generation. We create these benchmark fragments via the following strategy:

- **Functional Group**: A known functional group (e.g., `-OH`, `-NH2`, `-COOH`) is removed from the molecule. The model must regenerate the correct functional group given the remaining structure.
- **Heavy Atom**: Starting from a random atom in the graph, the last N heavy atoms (along with their hydrogens) in BFS traversal order are removed. The model

Table 3. Unconditional generation results on QM9. All models generate (almost exactly) 10,000 molecules. All metrics computed with `molmetrics`. PoseBusters sanity checks are computed on SMILES-valid molecules. Symphony++ was run with inverse temperatures: focus $= 1.0$, radial $= 1.0$, angular $= 20.0$.

	Validity (%)		Uniqueness (%)		Stability (%)		PoseBusters Sanity Checks (%)						
Model	`xyz2mol`	SMILES	`xyz2mol`	SMILES	Atom	Molecule	All Atoms Connected	Bond Lengths	Bond Angles	No Steric Clash	Aromatic Ring Flatness	Double Bond Flatness	Internal Energy
Symphony++	87.5	93.8	89.7	90.6	93.1	50.5	99.7	98.0	99.9	99.7	100.0	100.0	**98.5**
G-SphereNet	37.7	58.4	19.1	25.5	67.8	14.0	67.4	81.3	87.6	97.1	100.0	99.8	26.4
Symphony	77.4	83.8	97.8	98.1	90.7	42.0	98.5	97.8	99.8	98.9	100.0	100.0	95.9
GSchNet	78.1	90.7	96.7	97.4	95.7	68.0	99.0	99.4	99.9	99.0	100.0	99.9	95.0
EDM	84.5	95.4	**99.0**	**99.2**	98.4	81.8	99.7	**100.0**	99.9	99.8	100.0	100.0	96.0
GCDM	86.9	94.5	98.9	99.0	98.7	86.3	99.9	**100.0**	99.9	99.8	100.0	100.0	96.3
GeoLDM	91.3	95.0	98.8	98.9	**98.9**	**89.5**	99.7	**100.0**	**100.0**	99.5	100.0	100.0	96.8
ADiT	92.5	94.9	97.4	98.0	76.4	7.6	99.9	99.4	99.3	99.8	100.0	100.0	95.3
Zatom-1	92.0	94.9	97.0	97.2	98.2	84.9	**100.0**	**100.0**	99.9	99.8	100.0	100.0	97.3
SemlaFlow	**95.7**	**96.3**	97.5	97.5	97.1	78.5	**100.0**	**100.0**	**100.0**	**99.9**	100.0	100.0	97.1
QM9	*94.6*	*94.2*	*96.1*	*96.1*	*99.3*	*95.0*	*100.0*	*100.0*	*100.0*	*99.8*	*100.0*	*100.0*	*96.5*

Table 4. Symphony++ performance on the fragment completion benchmark on *unseen* QM9 test molecules. 100 completions were performed per mode. Inverse temperatures: focus $= 1.0$, radial $= 1.0$, angular $= 20.0$.

	Validity (%)		Completion Accuracy (%)			Stability (%)	
Mode	`xyz2mol`	SMILES	Exact Match	Same Formula	Same #Heavy	Atom	Molecule
Functional Group	99.0	100.0	48.0	48.0	69.0	99.1	91.0
Heavy Atom	98.0	100.0	24.0	29.0	76.0	97.3	79.0
Scaffold	96.0	100.0	13.0	17.0	66.0	98.2	85.0
Random Subgraph	90.0	94.0	1.0	4.0	64.0	94.7	58.0

must place them back correctly.

- **Scaffold**: The molecular scaffold (ring systems and linkers) is retained, but all substituent decorations are removed. The model must regenerate the substituents.
- **Random Subgraph**: A random connected subgraph is extracted as the fragment, and the model must complete the remainder. This is the most challenging mode since the missing atoms have no chemical prior.

To further test the generalization of Symphony++ and ensure that we are not simply testing memorization, we create these benchmark fragments using the *unseen* test split molecules. As seen in Table 4, Symphony++ samples valid completions for these molecular fragments, ranging from 90% validity in the hardest **Random Subgraph** setting to 99% validity in the easiest **Functional Group** setting. Encouragingly, the model often performs these valid completions in novel manners, showing that it has learned more general rules of chemistry.

4.3. Single Conditional QM9

We next show Symphony++'s capabilities for conditional molecular generation given a varying number of criteria. In this case, we condition on the quantum mechanical properties provided within QM9.

Like in the unconditional case, we sample from Symphony++ starting from a single `C` atom. For each property we condition on, we generate samples uniformly over a range of values centered at that property's mean within QM9. In addition to the previously described metrics, we evaluate our model's ability to satisfy the given conditions using an EGNN classifier for each property (as in (Hoogeboom et al., 2022) and (Satorras et al., 2022)), then use the resulting L1 loss as an additional metric.

Over the properties evaluated, Symphony++ performs comparably to, but slightly worse than, other conditional models (cG-SchNet (Gebauer et al., 2022) and GCLDM (Zhang et al., 2025)), as shown in Table 6. While `xyz2mol` validity is generally on par with the benchmarks, SMILES validity and uniqueness metrics are overall lower. On the other hand, Symphony++ consistently has the lowest classifier loss of the models compared, better satisfying the conditions given it than the other benchmarks (see Figure 3 for sample predictions).

4.4. Double Conditional QM9

We also evaluate Symphony++'s ability to satisfy two conditions simultaneously, using the same setup as in the single conditional setting. As in (Gebauer et al., 2022), we select the HOMO-LUMO gap and the "relative atomic energy" as defined in that previous work (i.e. a measure of whether the

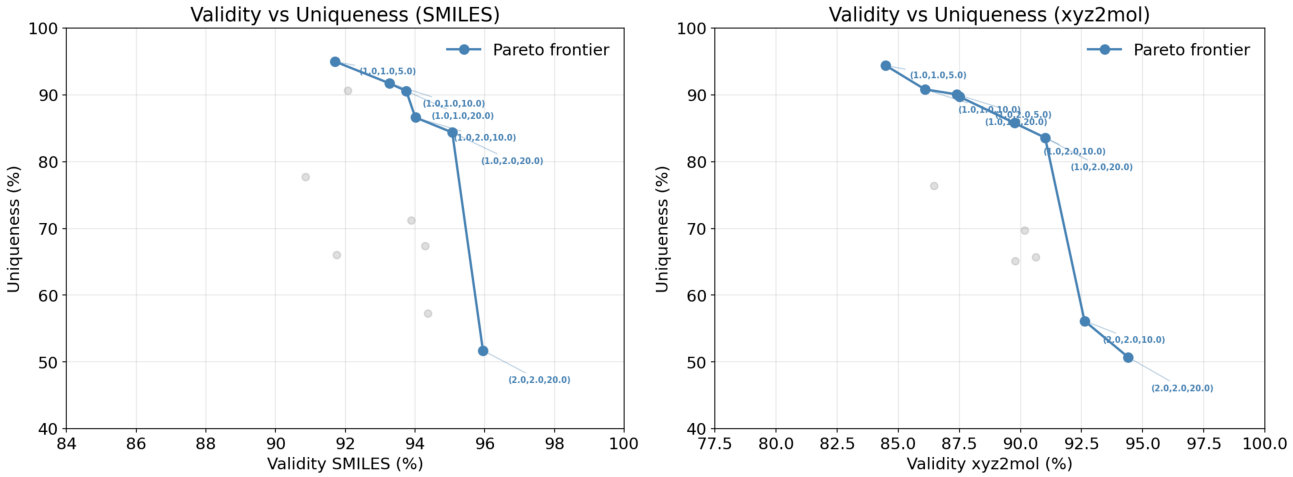

Figure 1. Validity (measured using SMILES and `xyz2mol`) against uniqueness for the unconditional QM9 model as evaluated at different inverse temperatures, as measured over 10000 samples starting from `C`. The Pareto frontiers are highlighted in blue. The labels indicate the focus, radial and angular inverse temperatures respectively.

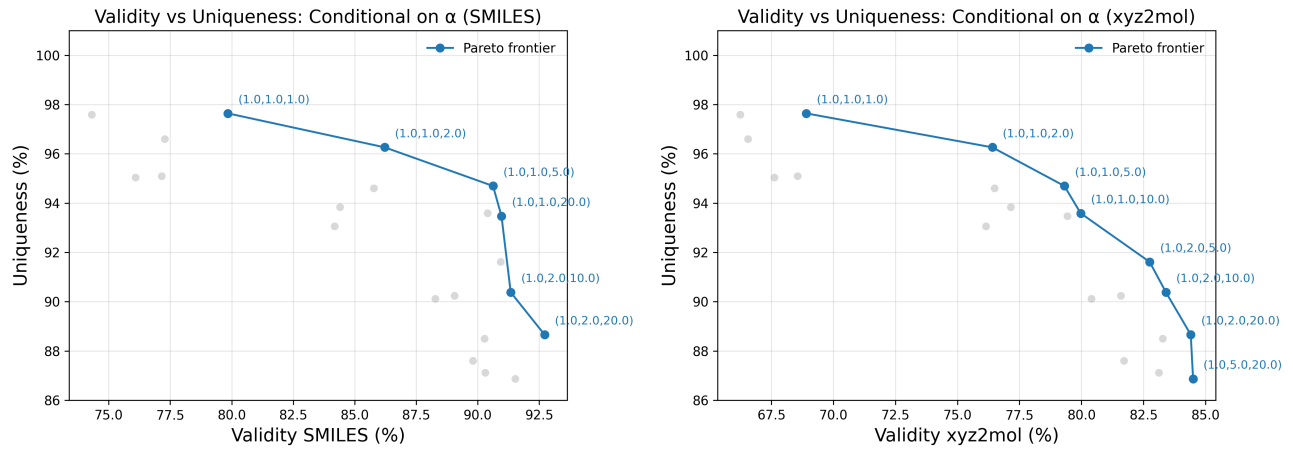

Figure 2. Validity (measured using SMILES and `xyz2mol`) against uniqueness for the conditional QM9 model, conditioned on polarizability, as evaluated at different angular inverse temperatures, measured over 20000 samples starting from `C`. he Pareto frontiers are highlighted in blue. The labels indicate the focus, radial and angular inverse temperatures respectively.

Table 5. Sampling metrics for Symphony++ at different inverse temperatures as measured over 5 starting elements (H, C, N, O, F) × 200 samples per starting element = 1000 samples in total. PoseBusters metrics are computed on valid molecules only.

Inverse Temperature			Validity (%)		Uniqueness (%)		Stability (%)		PoseBusters Sanity Checks (%)						
Focus	Radial	Angular	xyz2mol	SMILES	xyz2mol	SMILES	Atom	Molecule	All Atoms Connected	Bond Lengths	Bond Angles	No Steric Clash	Aromatic Ring Flatness	Double Bond Flatness	Internal Energy
1.0	1.0	10.0	87.6	94.0	97.9	98.2	90.6	43.6	99.6	97.6	99.9	99.7	100.0	100.0	98.5
1.0	1.0	20.0	87.1	94.0	**99.0**	**99.0**	90.9	42.7	99.3	98.7	99.8	99.5	100.0	100.0	99.4
1.0	2.0	20.0	89.6	94.9	96.9	97.2	93.0	53.6	99.8	97.8	100.0	99.3	100.0	100.0	99.3
2.0	2.0	20.0	**94.0**	**95.8**	83.9	85.0	**94.5**	**61.5**	**100.0**	**98.6**	**100.0**	**100.0**	100.0	100.0	**99.6**

Table 6. Per-property metrics across models conditioned on single properties. Evaluated on 22K structures per model. Metrics: Val (xyz) = Validity via xyz2mol (%), Val (SMI) = Validity via SMILES (%), Uniq. = Uniqueness (%), Loss = EDM Classifier Loss.

	Polarizability (α)				Heat Capacity (C_v)				Gap			
Model	Val (xyz)	Val (SMI)	Uniq.	Loss	Val (xyz)	Val (SMI)	Uniq.	Loss	Val (xyz)	Val (SMI)	Uniq.	Loss
Symphony++	**84.4**	92.7	88.7	**2.91**	73.5	80.6	70.1	**1.75**	80.2	87.6	70.8	**0.47**
cG-SchNet	77.0	**92.1**	**94.2**	30.97	**76.4**	**90.4**	96.6	3.05	72.3	88.1	**95.8**	0.97
GCLDM	63.5	75.7	82.1	11.07	16.6	75.8	**98.9**	16.85	**83.6**	**93.8**	92.1	1.06
	HOMO				**LUMO**				**Dipole Moment (μ)**			
Model	Val (xyz)	Val (SMI)	Uniq.	Loss	Val (xyz)	Val (SMI)	Uniq.	Loss	Val (xyz)	Val (SMI)	Uniq.	Loss
Symphony++	82.7	87.9	74.6	**0.30**	**72.4**	79.1	64.6	**0.46**	82.2	88.0	77.3	**0.75**
cG-SchNet	77.6	92.5	**96.6**	0.54	69.0	**83.2**	**95.5**	0.95	72.88	89.58	93.00	5.02
GCLDM	**87.5**	**94.8**	94.4	0.49	–	–	–	–	**85.6**	**93.5**	**94.2**	1.10

Table 7. Model performance when conditioned on HOMO-LUMO gap ("Gap") and relative atomic energy ("Rel. E.").

	Validity (%)		Uniq. (%)	Stability (%)		Loss	
Model	xyz2mol	SMILES		Atom	Molecule	Gap	Rel. E.
Symphony++	68.7	76.7	57.9	**93.7**	**73.7**	7.11	**0.046**
cG-SchNet	**68.9**	**84.0**	**64.2**	92.5	71.4	**6.52**	0.054

internal energy per atom is relatively high or low compared to other molecules of the same composition). We again evaluate over a selected distribution across both properties.

The multi-conditional setting is less commonly evaluated by other generative models, so we only compare our performance in this setting against cG-SchNet (Gebauer et al., 2022). While at least one other model we examine in this study also is theoretically capable of working with multiple conditions (GCLDM), it was not originally evaluated on this task and struggled to learn it properly in practice.

As shown in Table 7, Symphony++ and cG-SchNet perform very similarly in this setting, with cG-SchNet having slightly higher validity and Symphony++ having slightly better stability (after UFF relaxation). We show the distributions of each metric across the 2D gap/energy condition distribution in Figure 4 – model performance can vary greatly depending on where within the 2D distribution it is conditioned.

4.5. Steering Sampling for Non-Differentiable Objectives

Steering the sampling process for diffusion models has traditionally been done via classifier guidance (Dhariwal & Nichol, 2021), classifier-free guidance (Ho & Salimans, 2022), or inference time techniques such as Feynman-Kac steering (Singhal et al., 2025; Richman et al., 2026) which is closely related to Sequential Monte Carlo methods. These methods work well when the steering potential (or reward) is either known at training time or differentiable at inference time, allowing the gradient of the reward to update the transport process.

However, in practice, most rewards cannot be modelled in this way. This has led naturally to the use of reinforcement-learning (RL) based finetuning algorithms such as REINFORCE (Williams, 1992), PPO (Schulman et al., 2017) and GRPO (Shao et al., 2024) which have become very popular in the language modelling literature to optimize *non-differentiable* rewards in an unbiased manner using importance sampling. One of the key requirements for these algorithms to work well are unbiased estimates of model likelihoods for the samples it generates.

As we discussed in Section 3, the model likelihood estimates for each sampling step in Symphony++ are essentially free, unlike in the stochastic interpolant based baselines we compare to. (The probability estimates in these models can be approximated using the adjoint method which results in approximately 10× slower sampling speed (Klein et al., 2023).) By learning to optimize such rewards, the model

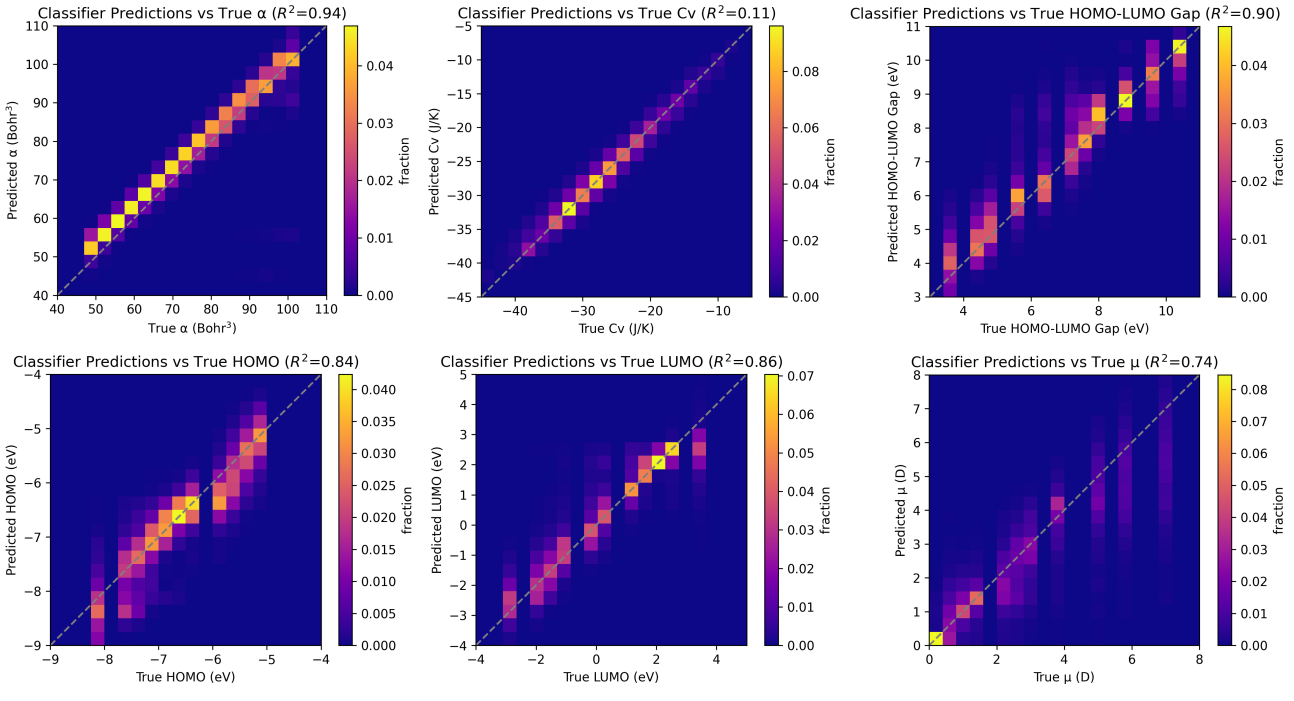

Figure 3. Predicted vs. true values of the QM9 properties that Symphony++ was conditioned on, as sampled along 11 discrete values per property (2000 samples per value).

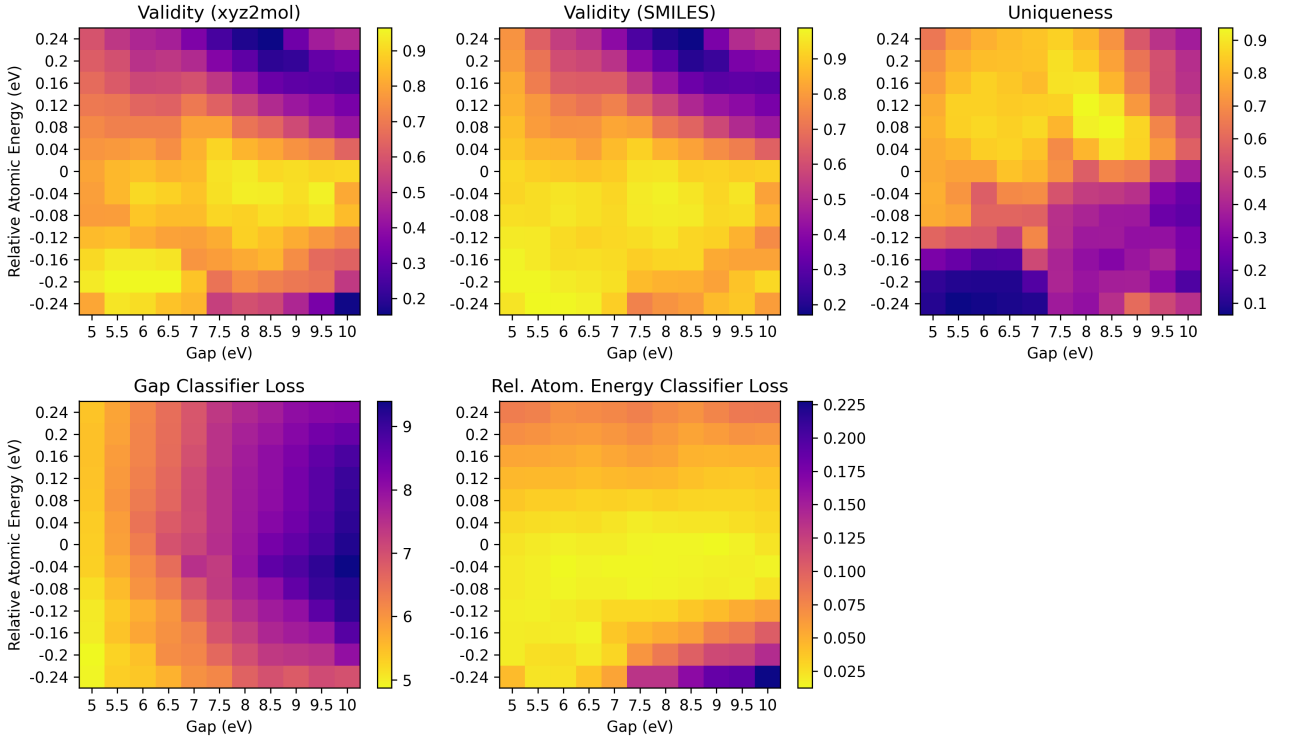

Figure 4. Metric heatmaps for double-conditioned Symphony++, evaluated over varying values of HOMO-LUMO gap and relative atomic energy (2000 samples per (gap, energy) pair).

potentially learns to sample beyond its training distribution. This has historically been impossible for the baselines we compare to, due to both their slow sampling speed and lack of easy-to-compute unbiased likelihood estimates. Further, since the intermediate steps of the sampling process in the stochastic interpolant models do not usually correspond to realistic molecules, the reward can only be reasonably estimated at the final sample. Thus, the entire sampling process (which can be anywhere from 100 to 500 discretized steps irrespective of the size of the final molecule) has to be differentiated through, which is prohibitively expensive. In the case of Symphony++, reward shaping to guide intermediate steps of the sampling process is a definite possibility. We leave the application of RL-based finetuning to Symphony++ for future work.

5. Conclusion

We have developed Symphony++, a new autoregressive model improving on the recipe from Daigavane et al. (2024). While Symphony++ still lags state-of-the-art models on traditional validity and uniqueness metrics in the unconditional setting, it is extremely efficient to train and sample, and allows for flexible conditioning on both geometrical and quantum mechanical properties, outperforming existing models in the conditional setting with respect to matching the conditions provided. Importantly, this arises from the model's ability to both take as input partially completed molecular fragments as well as output variably sized molecules. In the future, we plan to leverage Symphony++'s fast sampling and exact likelihoods for reinforcement-learning based finetuning to optimize non-differentiable rewards.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A. J., Bambrick, J., Bodenstein, S. W., Evans, D. A., Hung, C.-C., O'Neill, M., Reiman, D., Tunyasuvunakool, K., Wu, Z., Žemgulytė, A., Arvaniti, E., Beattie, C., Bertolli, O., Bridgland, A., Cherepanov, A., Congreve, M., Cowen-Rivers, A. I., Cowie, A., Figurnov, M., Fuchs, F. B., Gladman, H., Jain, R., Khan, Y. A., Low, C. M. R., Perlin, K., Potapenko, A., Savy, P., Singh, S., Stecula, A., Thillaisundaram, A., Tong, C., Yakneen, S., Zhong, E. D., Zielinski, M., Žídek, A., Bapst, V., Kohli, P., Jaderberg, M., Hassabis, D., and Jumper, J. M. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, 2024.

Albergo, M. S., Boffi, N. M., and Vanden-Eijnden, E. Stochastic interpolants: A unifying framework for flows and diffusions, 2023.

Anstine, D. M. and Isayev, O. Generative models as an emerging paradigm in the chemical sciences. *Journal of the American Chemical Society*, 145(16):8736–8750, 04 2023.

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

Batzner, S., Musaelian, A., Sun, L., Geiger, M., Mailoa, J. P., Kornbluth, M., Molinari, N., Smidt, T. E., and Kozinsky, B. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. 13, May 2022. URL https://doi.org/10.1038/s41467-022-29939-5.

Billera, L., Nordlinder, H. N., Ryder, J. C., Oresten, A., Stålmarck, A., Björk, T. M., and Murrell, B. Branching flows: Discrete, continuous, and manifold flow matching with splits and deletions, 2026. URL https://arxiv.org/abs/2511.09465.

Buttenschoen, M., Morris, G. M., and Deane, C. M. PoseBusters: AI-based docking methods fail to generate physically valid poses or generalise to novel sequences, 2023.

Cheng, A. H., Sun, C., and Aspuru-Guzik, A. Scalable autoregressive 3d molecule generation, 2025. URL https://arxiv.org/abs/2505.13791.

Daigavane, A., Ravindran, B., and Aggarwal, G. Understanding Convolutions on Graphs. *Distill*, 2021. doi: 10.23915/distill.00032. https://distill.pub/2021/understanding-gnns.

Daigavane, A., Kim, S. E., Geiger, M., and Smidt, T. Symphony: Symmetry-equivariant point-centered spherical harmonics for 3d molecule generation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=MIEnYtlGyv.

Dao, T., Fu, D. Y., Ermon, S., Rudra, A., and Ré, C. Flashattention: Fast and memory-efficient exact attention with io-awareness, 2022. URL https://arxiv.org/abs/2205.14135.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.),

Advances in Neural Information Processing Systems, volume 34, pp. 8780–8794. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/49ad23d1ec9fa4bd8d77d02681df5cfa-Paper.pdf.

Dunn, I. and Koes, D. R. Flowmol3: Flow matching for 3d de novo small-molecule generation, 2025. URL https://arxiv.org/abs/2508.12629.

Gebauer, N., Gastegger, M., and Schütt, K. Symmetry-adapted generation of 3d point sets for the targeted discovery of molecules. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/a4d8e2a7e0d0c102339f97716d2fdfb6-Paper.pdf.

Gebauer, N. W. A., Gastegger, M., Hessmann, S. S. P., Müller, K.-R., and Schütt, K. T. Inverse design of 3d molecular structures with conditional generative neural networks. 13, February 2022. URL https://doi.org/10.1038/s41467-022-28526-y.

Geffner, T., Didi, K., Cao, Z., Reidenbach, D., Zhang, Z., Dallago, C., Kucukbenli, E., Kreis, K., and Vahdat, A. La-proteina: Atomistic protein generation via partially latent flow matching, 2025. URL https://arxiv.org/abs/2507.09466.

Geiger, M., Kucukbenli, E., Zandstein, B., and Tretina, K. Accelerate drug and material discovery with new math library NVIDIA cuEquivariance, 11 2024. URL https://developer.nvidia.com/blog/accelerate-drug-and-material-discovery-with-new-math-library-nvidia-cuequivariance/. NVIDIA Developer Blog.

Havasi, M., Karrer, B., Gat, I., and Chen, R. T. Q. Edit flows: Flow matching with edit operations, 2025. URL https://arxiv.org/abs/2506.09018.

Ho, J. and Salimans, T. Classifier-free diffusion guidance, 2022. URL https://arxiv.org/abs/2207.12598.

Holderrieth, P., Albergo, M. S., and Jaakkola, T. Leaps: A discrete neural sampler via locally equivariant networks, 2025. URL https://arxiv.org/abs/2502.10843.

Hoogeboom, E., Satorras, V. G., Vignac, C., and Welling, M. Equivariant Diffusion for Molecule Generation in 3D, 2022.

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

Joshi, C. K., Fu, X., Liao, Y.-L., Gharakhanyan, V., Miller, B. K., Sriram, A., and Ulissi, Z. W. All-atom diffusion transformers: Unified generative modelling of molecules and materials, 2025. URL https://arxiv.org/abs/2503.03965.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., and Hassabis, D. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Kim, Y. and Kim, W. Y. Universal Structure Conversion Method for Organic Molecules: From Atomic Connectivity to Three-Dimensional Geometry. *Bulletin of the Korean Chemical Society*, 36(7): 1769–1777, 2015. doi: https://doi.org/10.1002/bkcs.10334. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/bkcs.10334.

Kingma, D. P. and Welling, M. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, November 2019. ISSN 1935-8245. doi: 10.1561/2200000056. URL http://dx.doi.org/10.1561/2200000056.

Klein, L., Foong, A. Y. K., Fjelde, T. E., Mlodozeniec, B., Brockschmidt, M., Nowozin, S., Noé, F., and Tomioka, R. Timewarp: Transferable acceleration of molecular dynamics by learning time-coarsened dynamics, 2023. URL https://arxiv.org/abs/2302.01170.

Kleinhenz, J. and Daigavane, A. e3tools, April 2025.

Koker, T., Kotak, M., and Smidt, T. Training a foundation model for materials on a budget. *arXiv preprint arXiv:2508.16067*, 2025.

Liao, Y.-L., Wood, B., Das, A., and Smidt, T. Equiformerv2: Improved equivariant transformer for scaling to higher-degree representations, 2024. URL https://arxiv.org/abs/2306.12059.

Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling, 2023.

Luo, Y. and Ji, S. An Autoregressive Flow Model for 3D Molecular Geometry Generation from Scratch. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=C03Ajc-NS5W.

Makurvet, F. D. Biologics vs. small molecules: Drug costs and patient access. *Medicine in Drug Discovery*, 9: 100075, 2021.

Morehead, A., Cretu, M., Panescu, A., Anand, R., Weiler, M., Perez, T., Blau, S., Farrell, S., Bhimji, W., Jain, A., Sahasrabuddhe, H., Lio, P., Jaakkola, T., Gomez-Bombarelli, R., Ying, R., Erichson, N. B., and Mahoney, M. W. Zatom-1: A multimodal flow foundation model for 3d molecules and materials, 2026. URL https://arxiv.org/abs/2602.22251.

Nikitin, F., Dunn, I., Koes, D. R., and Isayev, O. Geomdrugs revisited: toward more chemically accurate benchmarks for 3d molecule generation. *Digital Discovery*, 4: 3282–3291, 2025.

Potaptchik, P., Yim, J., Saravanan, A., Holderrieth, P., Vanden-Eijnden, E., and Albergo, M. S. Discrete flow maps, 2026. URL https://arxiv.org/abs/2604.09784.

Ramakrishnan, R., Dral, P. O., Rupp, M., and von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1(1):140022, 2014.

Richman, D. D., Karaguesian, J., Suomivuori, C.-M., and Dror, R. O. Unlocking hidden biomolecular conformational landscapes in diffusion models at inference time, 2026. URL https://arxiv.org/abs/2512.03312.

Satorras, V. G., Hoogeboom, E., and Welling, M. E(n) Equivariant Graph Neural Networks, 2022.

Schneuing, A., Igashov, I., Dobbelstein, A. W., Castiglione, T., Bronstein, M., and Correia, B. Multi-domain distribution learning for de novo drug design, 2025. URL https://arxiv.org/abs/2508.17815.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms, 2017.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y. K., Wu, Y., and Guo, D. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Simm, G., Pinsler, R., and Hernandez-Lobato, J. M. Reinforcement learning for molecular design guided by quantum mechanics. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8959–8969. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/simm20b.html.

Simm, G. N. C., Pinsler, R., Csányi, G., and Hernández-Lobato, J. M. Symmetry-aware actor-critic for 3d molecular design. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=jEYKjPE1xYN.

Singhal, R., Horvitz, Z., Teehan, R., Ren, M., Yu, Z., McKeown, K., and Ranganath, R. A general framework for inference-time scaling and steering of diffusion models, 2025. URL https://arxiv.org/abs/2501.06848.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations, 2021.

Stark, H., Jing, B., Wang, C., Corso, G., Berger, B., Barzilay, R., and Jaakkola, T. Dirichlet flow matching with applications to dna sequence design, 2024. URL https://arxiv.org/abs/2402.05841.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3–4):229–256, May 1992. ISSN 0885-6125. doi: 10.1007/BF00992696. URL https://doi.org/10.1007/BF00992696.

Zhang, Q., Xiao, J., Niu, D., Zhang, Z., Ding, S., and Li, Z. Geometry-complete latent diffusion model for 3d molecule generation. *Bioinformatics*, 41(8): btaf426, 08 2025. ISSN 1367-4811. doi: 10.1093/bioinformatics/btaf426. URL https://doi.org/10.1093/bioinformatics/btaf426.