InertialTransformer: Early Explorations and Insights into Transformer-based Geometric Representation

Haorui Li[1], **Weitao Du**[2], **Hongyu Guo**[3], **Shengchao Liu**[1]
[1]The Chinese University of Hong Kong
[2]Alibaba DAMO Academy
[3]University of Ottawa
scliu@cse.cuhk.edu.hk

Abstract

In many biochemical studies, molecular geometries serve as fundamental data structures. Existing deep learning methods often focus on designing SE(3)-equivariant representation functions. However, such functions are physically constrained, which may limit the expressiveness of the models. In this work, we introduce InertialTransformer, a preliminary attempt to address this challenge. InertialTransformer comprises three key components: (1) it uses the inertial frame as a canonicalization method to align molecular geometries in 3D Euclidean space; (2) it incorporates a Euclidean-based positional encoding scheme; and (3) it employs a self-attention module to enable information exchange among atoms. By integrating these components, InertialTransformer achieves an SE(3)-equivariant yet unconstrained framework for geometric representation. We evaluate InertialTransformer on molecular geometry prediction tasks. While its performance does not yet match that of state-of-the-art 3D graph neural networks, it significantly outperforms existing SE(3)-equivariant Transformer-based approaches. We posit that InertialTransformer stands to benefit substantially from large-scale pretraining, which we leave as a direction for future work.

1 Introduction

Transformer architectures have had a profound impact on natural language processing (Devlin et al., 2019; Radford et al., 2019) and computer vision (Dosovitskiy et al., 2020; Khan et al., 2022), due to their pioneering self-attention mechanism and positional encoding (Vaswani, 2017; Su et al., 2024). The key strengths of Transformer lie in its ability to effectively capture global contextual information across the entire input tokens while exhibiting superior scalability on large-scale datasets. These characteristics have established Transformers as the dominant architecture for developing state-of-the-art foundation models (Achiam et al., 2023; Liu et al., 2024).

The emergence of "AI for Science" has catalyzed significant research efforts to harness the Transformer's capabilities for scientific domains, particularly in chemistry, materials science, and biology (Li et al., 2024; Flam-Shepherd & Aspuru-Guzik, 2023; Yan et al., 2024; Fu et al., 2024). Among these scientific tasks, the molecular geometries serve as the fundamental data structures; thus, how to incorporate the physical constraints, *i.e.*, SE(3)-equivariance, into the modeling is a critical challenge.

Along this line, existing methods are SE(3)-constrained functions and can be roughly divided into two venues: graph neural network (GNN)-based and Transformer-based. They have shown remarkable performance, including but not limited to 3D molecular property prediction (Liu et al., 2023; Schütt et al., 2021; Shi et al., 2023) and large-scale geometric self-supervised pretraining (Liu et al., 2021). The connection between Transformers and GNNs lies in the attention mechanism, which can be interpreted as a GNN layer applied to a fully connected graph, where each node corresponds to a token. The key distinction is that Transformers incorporate explicit positional encoding and tokenization modules,

which are typically absent in standard GNNs. In this work, we focus on Transformers, which have emerged as one of the most expressive and widely adopted architectures in the AI community (Touvron et al., 2023; Bai et al., 2023; Minaee et al., 2025). Notably, unlike conventional sequential data (*e.g.*, texts in NLP) or grid-based data (*e.g.*, images in CV), molecular geometries, as a set of atoms in the 3D Euclidean space, pose unique challenges for the tokenization, as will be discussed next.

Physical Constraint and Model Expressiveness. The molecular representations are expected to preserve SE(3)-equivariance, *i.e.*, the output transforming correspondingly under rotations and translations of the whole molecular system (Thomas et al., 2018). Most existing methods achieve this by designing physics-constrained neural networks (Liu et al., 2023). However, another research line proposes an alternative solution by simply performing data augmentation without requiring the model to be SE(3)-equivariant (Abramson et al., 2024). The underlying conjecture is that without enforcing the physical constraint, the deep learning models can possess stronger model expressiveness, thus leading to better model performance. Motivated by this observation, we raise a critical research question: *Would it be possible to design a Transformer method, such that it can satisfy the physical constraint while possessing strong model expressiveness?*

Our Contributions. To answer this question, in this work, we introduce the InertialTransformer, a Transformer-based architecture targeting at solving the structure tokenization. As shown in Figure 1, there are three key steps in InertialTransformer. (1) The first key step in InertialTransformer is the utility of the inertial frame for pose canonicalization. More concretely, we first fix the molecular system to its center of mass and then align it with its inertial frame, resulting in an invariant canonical pose. (2) The second key step is to introduce a structure-aware tokenization strategy, where the goal is to encode all the geometric information, *i.e.*, scalars and vectors, into the positional encodings. (3) Last but not least, these encodings are designed to carry sufficient information to be effectively utilized by the self-attention layers.

Results and Limitations. To verify the effectiveness of the InertialTransformer, we conduct experiments on molecular 3D property prediction tasks. By comparing with relevant Transformer-based and GNN-based baselines, we get the following interesting observations. Along the research line of 3D Transformer with structure tokenization, InertialTransformer demonstrates superior performance compared to the existing baseline (Li et al., 2024). However, when evaluated against 3D GNN baselines, while our model surpasses certain established approaches, it maintains competitive performance without surpassing current state-of-the-art 3D GNN architectures. We posit that a large-scale pretraining could address this limitation, and leave this as the next step. We openly acknowledge these unresolved questions and share these observations with the community in the hope of fostering discussion. We welcome any constructive feedback or suggestions.

2 Related Works

Geometric Representation Learning for 3D Molecules. Learning effective representations of 3D molecular structures requires handling SE(3) symmetry—ensuring representations remain invariant or equivariant under rotations and translations. Current approaches can be categorized into four main paradigms. SE(3)-equivariant architectures explicitly enforce symmetry through specialized network designs: spherical frame basis models (Thomas et al., 2018; Liao & Smidt, 2023) project features into irreducible representations of SO(3), while vector frame basis models (Schütt et al., 2021; Satorras et al., 2022) construct local coordinate frames for equivariant operations. Invariant feature approaches circumvent architectural constraints by utilizing geometrically invariant inputs such as pairwise distances, bond angles, and dihedral angles (Schütt et al., 2017; Gasteiger et al., 2022). Data augmentation strategies encourage models to implicitly learn symmetric representations by training on randomly rotated and translated molecular conformations, particularly valuable for large-scale models where explicit equivariance is complex to scale (Flam-Shepherd & Aspuru-Guzik, 2023; Abramson et al., 2024). Input canonicalization methods (Antunes et al., 2024; Yan et al., 2024; Li et al., 2024; Fu et al., 2024) establish a canonical orientation or

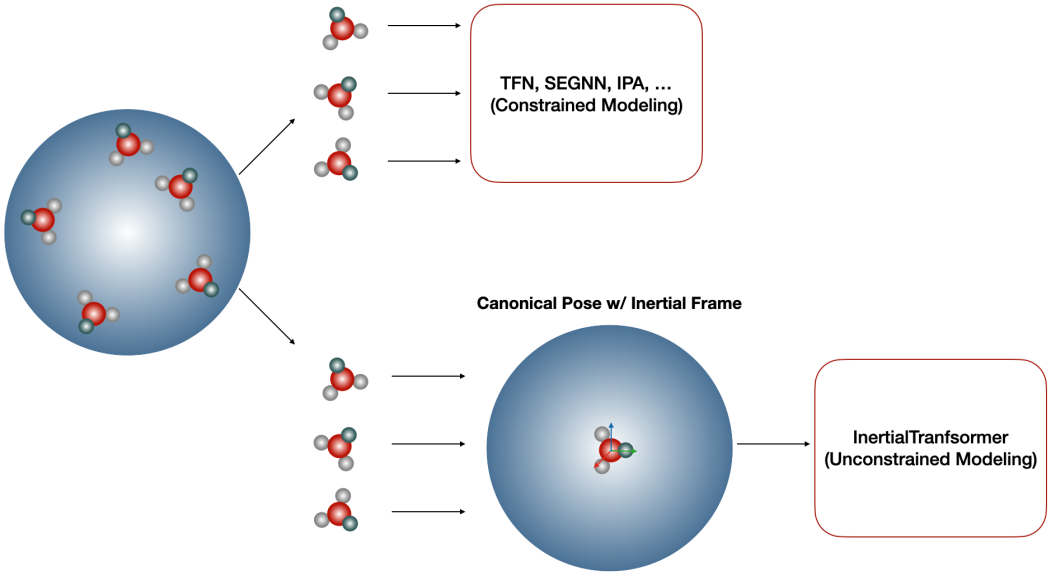

Figure 1: Comparison of existing SE(3)-equivariant graph neural networks and InertialTransformer.

reference frame for input molecules through preprocessing, transforming each molecule into a standardized pose so that subsequent neural networks can operate on SE(3)-invariant inputs without intrinsic SE(3)-equivariant constraints.

Transformer-based Geometric Modeling of 3D Molecules. The adaptation of Transformer architectures for 3D molecular modeling necessitates strategies to incorporate geometric information and handle SE(3) symmetries. Fuchs et al. (2020) develops equivariant architectures that integrate self-attention principles with Graph Neural Networks, emphasizing local neighborhood interactions and intricate geometric feature transformations. Shi et al. (2023) incorporates SE(3)-invariant geometric features, such as pairwise inter-atomic distances and structural descriptors, as bias terms in Transformer attention computation. This enables enhanced capture of spatial relationships without requiring strict architectural equivariance constraints. While these two approaches ensure strong geometric inductive biases, they **actually operate as a message-passing GNN** and do not fully leverage the parallel, all-to-all attention mechanisms of standard Transformers. Li et al. (2024) addresses SE(3) properties through preprocessing molecular input data rather than explicit architectural constraints. Under this research line, approaches typically perform data augmentation or input canonicalization, and then convert 3D molecular structures into specialized tokenized representations, which is crucial for standard Transformer architectures.

Structure Tokenization of 3D Molecules. The discretization of 3D molecular structures into tokenized representations compatible with Transformer models poses significant technical challenges. Text sequence-based tokenization (Li et al., 2024; Yan et al., 2024; Flam-Shepherd & Aspuru-Guzik, 2023) directly converts 3D molecular structures into 1D "textualized" sequences. This approach treats molecular structures similarly to natural language text by directly tokenizing each atom's type and its continuous 3D coordinates into discrete sequential tokens. However, current tokenization methods still face challenges in generating tokenized representations that preserve complete three-dimensional geometric information while maintaining compatibility with standard Transformer architectures, indicating promising directions for future exploration.

3 InertialTransformer

In this section, we will go over the detailed design of InertialTransformer. In Section 3.1, we introduce an inertial frame as an SE(3)-equivariant canonicalization method, leading to an in-

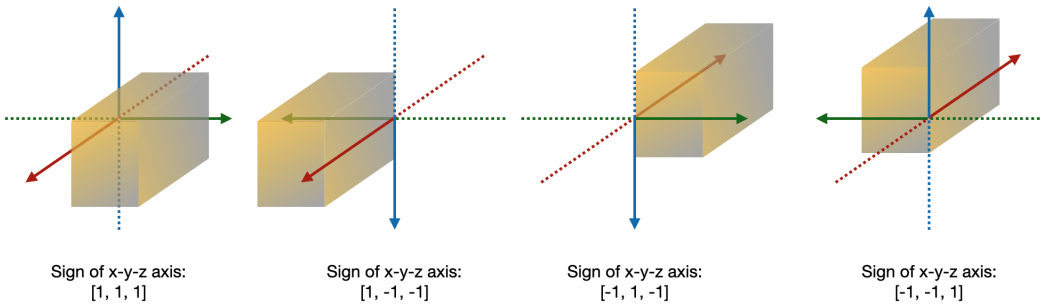

Figure 2: Illustration of introducing a fourth node as the anchor node. We define the sign of the x-y-z axis to make sure that $\boldsymbol{x}_4$ is in the first quadrant, and there are four cases as illustrated in the four subfigures, respectively.

variant structure for each molecular system. Following this, we establish a Transformer-like framework, specifically the expressive geometric-aware self-attention module, in Section 3.2. Notice that in addition to the attention module, tokenization is another important module. The interesting part in InertialTransformer is how to design the geometrically aware tokenization on top of the inertial frame, as will be introduced in Section 3.3.

3.1 Inertial Frame Construction

First, we employ the following four sequential steps to derive the reference frames that construct the rotation matrix from N atomic positions $\boldsymbol{r}$:

- Calculate the mass center: $\boldsymbol{c} = \frac{1}{N}\sum_i \boldsymbol{r}_i$.
- Adjust position relative to the center $\boldsymbol{r}_i = \boldsymbol{r}_i - \boldsymbol{c}$.
- Compute the inertia tensor $\hat{I} = \sum_i \|\boldsymbol{r}_i\|^2 I - \boldsymbol{r}_i \boldsymbol{r}_i^T$, where I is the unit diagonal matrix.
- Obtain the principal axes of inertia by applying eigen-decomposition on $\hat{I}$. We have $\hat{I} = Q\Lambda Q^T$, where Q is the orthogonal matrix whose columns are the eigenvectors of $\hat{I}$, and Λ is the diagonal matrix whose elements are the eigenvalues λ_i of $\hat{I}$, representing the principal moments of inertial along the principal axes.

How to define the orderings of inertial frame axes? We follow the ordering of the eigenvalues to define the ordering of the eigen-vectors, which form the rotation matrix. The key point to note is how to handle the tie between the eigenvalues. In such cases, the molecular system is symmetric (*e.g.*, CO_2 or CH_4), leading to degenerate eigenvalues of the inertia tensor. Consequently, the inertial frame is not uniquely defined, yet all valid frames are physically equivalent.

How to define the directions of inertial frame axes? The orthonormal I is the basis. Meanwhile, there are eight possible combinations for the directions or signs of the x-, y-, and z-axes, given by $\{\pm 1, \pm 1, \pm 1\}$, respectively. First, we enforce the ordering of the x-y-z axis to be right-handed, *i.e.*, the determinant of I to be 1, not -1. This still gives us four possible combinations. Then we can define a unique direction for each molecule system by introducing a fourth node, as in Theorem 1.

Theorem 1. *For an inertial frame F, we build up the corresponding right-handed axes as coordinate systems Q. Then we need to incorporate a fourth point that is not on the y-z plane or x-z plane to uniquely determine the directions of the coordinate system with one rotation transformation matrix.*

As illustrated in Theorem 1, we must include a fourth node to uniquely determine the directions of the three axes. To achieve this, we consider a fourth node $\boldsymbol{x}_4$ that is not on the y-z plane or x-z plane and with the largest distance to the origin. Then we define the *requirement* that x-$\boldsymbol{x}_4$-z and $\boldsymbol{x}_4$-y-z are also right-handed; in other words, this requirement is essentially saying that $\boldsymbol{x}_4$ should be in the first quadrant of the x-y plane. For implementation, $\boldsymbol{x}_4$ is a 3D point whose projection onto the x–y plane falls into one of the four quadrants: the first, second, third, or fourth quadrant, depending on the signs of its x and y coordinates.

Each of them defines the signs (or directions) of the inertial frame axes, as illustrated in Figure 2.

Summary. All these options for finding an inertial frame for each molecule are to find a canonical pose by rotating the molecule system to have the inertial frame lie precisely in the x-y-z axes, as shown in Figure 1. The expectation is that such a canonical posing of molecules can (1) serve as an invariant global frame system for follow-up modules, (2) naturally merge with the Transformer architecture, and (3) support stronger generation ability.

3.2 Method: InertialTransformer

In Section 3.1, we introduce the use of an inertial frame as a global frame for each molecule. We rotate the entire molecular system into a canonical pose, aligning its inertial frame with the global x–y–z axes.

Building upon this canonical data structure, we develop InertialTransformer, which follows the standard Transformer architecture with three key components: tokenization, positional encoding, and multi-head self-attention. (1) Structure-aware tokenization encodes both the discrete atomic type and continuous Euclidean coordinates for each atom into a unified token representation. (2) Geometric positional encoding addresses the inherent position-agnostic nature of Transformer architectures by incorporating spatial relationship information. We employ a specialized positional encoding scheme designed for continuous Euclidean coordinates, as detailed in Section 3.3. (3) The encoded tokens are subsequently processed through multi-head self-attention mechanisms (Vaswani, 2017), enabling the model to capture inter-atomic relationships and dependencies.

The core insight of our approach is that through appropriate positional encoding, pairwise atomic relationships can be effectively learned via inner product operations between token representations within the attention mechanism. Here we discuss two types of tokenization methods, and then employ the standard self-attention mechanism, as shown in Equation (1).

$$a = \frac{K^T Q}{\sqrt{d}}, \qquad o = aV. \tag{1}$$

3.3 Positional Encoding for InertialTransformer

3.3.1 RoPE as Separate Tokenization

What are we expecting for the position embedding? We hope to design a position embedding such that it can hold both the absolute position while maintaining the relative distance when calculating the inner product, *i.e.*, $R^T_{x_1,y_1,z_1} R_{x_2,y_2,z_2} = R_{x_2-x_1,y_2-y_1,z_2-z_1}$.

Inspired by Su (2021), we propose the geometric position embedding for atomic points in the Euclidean space:

$$R_{x,y,z}\boldsymbol{q} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \end{bmatrix} \cdot \begin{bmatrix} \cos x\theta_0 \\ \cos x\theta_0 \\ \cos y\theta_0 \\ \cos y\theta_0 \\ \cos z\theta_0 \\ \cos z\theta_0 \end{bmatrix} + \begin{bmatrix} -q_1 \\ q_0 \\ -q_3 \\ q_2 \\ -q_5 \\ q_4 \end{bmatrix} \cdot \begin{bmatrix} \sin x\theta_0 \\ \sin x\theta_0 \\ \sin y\theta_0 \\ \sin y\theta_0 \\ \sin z\theta_0 \\ \sin z\theta_0 \end{bmatrix}. \tag{2}$$

By this, we can tell that $\boldsymbol{q}^T R^T_{x_1,y_1,z_1} R_{x_2,y_2,z_2} \boldsymbol{k} = \boldsymbol{q}^T R_{x_2-x_1,y_2-y_1,z_2-z_1} \boldsymbol{k}$.

3.3.2 Nyström Approximation for Pairwise Distance Tokenization

One limitation of using RoPE-3D in Equation (2) for structure tokenization is that it treats each axis separately. Though by expectation, it should be able to learn the token pairwise distance information. We empirically observe that merely using RoPE-3D cannot learn adequate information, while explicitly adding the pairwise information is more informative.

Then the question is how to incorporate the pairwise distance into the model. One straightforward way is to directly inject the distance information into the attention score, like Shi et al. (2023). However, such an architecture is not compatible with the classic transformer architecture used in large language models (Bai et al., 2023; Achiam et al., 2023; Touvron et al., 2023), which is not suitable for us because the ultimate goal along this research line is to enable the multi-modal alignment between the geometric structures and other modalities like natural language.

To alleviate this issue, we consider the Nyström method (Williams & Seeger, 2000). It is a low-rank approximation to obtain the pairwise distance. More concretely, suppose we have a Gram matrix over n points, *i.e.*, $K \in \mathbb{R}^{n \times n}$. Each element K_{ij} is the radial basis function (RBF) over the distance between i-th and j-th points, $K_{ij} = RBF(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$. Then we sample m anchor points, $(\boldsymbol{c}_1, \boldsymbol{c}_2, ..., \boldsymbol{c}_m)$, where each is a 3D position in an Euclidean space and $n \gg m$.

First we can decompose matrix K with eigendecomposition,

$$K = U \Lambda U^T, \tag{3}$$

where $U \in \mathbb{R}^{n \times n}$ is an orthonormal matrix with eigenvecturos, and $\Lambda \in \mathbb{R}^{m \times m}$ is block diagonal matrix with eigenvalues. Then, Nyström approximation is a low rank approximation, assuming that matrix K can be approximated using $\tilde{K}$:

$$\begin{aligned} K &\approx \tilde{K} \\ &= \tilde{U} \tilde{\Lambda} \tilde{U}^T \\ &= \begin{bmatrix} A & B \\ B^T & C \end{bmatrix}, \end{aligned} \tag{4}$$

where $\tilde{U}$ is the first m columns of U and $\tilde{\Lambda}$ is the block diagonal matrix of first m eigenvalues of Λ. At this point, we assume that the m points picked can estimate the m-rank matrix A with positive eigenvalues. Then let us have $\tilde{U} = \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}$, where $U_1 \in \mathbb{R}^{m \times m}$ and $U_2 \in \mathbb{R}^{(n-m) \times m}$, and $A = U_1 \tilde{\Lambda} U_1^T$. Thus, we can rewrite Equation (4) as:

$$\begin{aligned} \tilde{K} &= \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} \tilde{\Lambda} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}^T \\ &= \begin{bmatrix} U_1 \Lambda U_1^T & U_1 \Lambda U_2^T \\ U_2 \Lambda U_1^T & U_2 \Lambda U_2^T \end{bmatrix}. \end{aligned} \tag{5}$$

Combining this with Equation (4), we have $U_2 = B^T U_1 \Lambda^{-1}$ and $U_2^T = \Lambda^{-1} U_1^T B$. Thus, we can have

$$C = U_2 \Lambda U_2^T = B^T U_1 \Lambda^{-1} U_1^T B = B^T A^{-1} B. \tag{6}$$

To inject this back to Equation (4), we have

$$\begin{aligned} \tilde{K} &= \tilde{U} \tilde{\Lambda} \tilde{U}^T \\ &= \begin{bmatrix} U \\ B^T U \Lambda^{-1} \end{bmatrix} \Lambda \begin{bmatrix} U^T & \Lambda^{-1} U^T B \end{bmatrix} \\ &= \begin{bmatrix} A & B \\ B^T & B^T A^{-1} B \end{bmatrix} \\ &= \begin{bmatrix} A \\ B^T \end{bmatrix} A^{-1} \begin{bmatrix} A & B \end{bmatrix}. \end{aligned} \tag{7}$$

This wraps up the key idea of Nyström method. Then, to obtain the RBF of a new point pair $K(i, j)$, we first construct the feature between point i, j and the m anchor points as $k_i =$

Featurization	Model	$\alpha \downarrow$ a_0^3	$\nabla\mathcal{E} \downarrow$ meV	$\mathcal{E}_{\text{HOMO}} \downarrow$ meV	$\mathcal{E}_{\text{LUMO}} \downarrow$ meV	$\mu \downarrow$ D	$C_v \downarrow$ $\frac{cal}{mol \cdot K}$	$G \downarrow$ meV	$H \downarrow$ meV	$R^2 \downarrow$ a_0^2	$U \downarrow$ meV	$U_0 \downarrow$ meV	ZPVE $\downarrow$ meV
1D FPs	MLP	2.231	196.72	131.27	164.94	0.526	0.919	2158.64	2358.23	68.621	2340.61	2314.77	155.921
	RF	3.801	207.02	165.72	183.04	0.534	1.485	3391.79	3729.94	94.512	3705.75	3678.25	253.132
	XGB	2.748	199.71	139.88	165.43	0.516	1.062	2563.93	2804.27	82.959	2786.28	2769.29	180.989
1D SMILES	CNN	0.364	165.22	124.65	114.81	0.566	0.173	156.66	170.59	20.403	166.18	169.89	10.070
	BERT	0.313	117.50	84.93	98.88	0.446	0.176	170.01	183.43	18.002	183.84	188.60	13.410
1D SELFIES	CNN	0.345	157.04	115.51	113.00	0.499	0.168	136.42	146.56	20.080	143.00	140.01	10.149
	BERT	0.348	123.11	91.15	90.80	0.461	0.203	168.20	187.50	19.125	204.93	195.98	17.328
2D Graph	GCN	1.338	145.82	96.21	106.66	0.434	0.526	1198.12	1291.57	37.585	1281.03	1303.39	85.103
	ENN-S2S	1.401	270.59	129.18	132.84	0.577	0.760	1487.21	955.24	34.609	1800.79	1521.32	51.226
	GraphSAGE	1.601	131.45	88.78	93.21	0.402	0.544	1473.42	1617.73	38.112	1553.01	1565.65	95.344
	GAT	1.132	135.90	94.70	98.52	0.406	0.291	911.82	991.31	26.583	1161.29	592.67	55.061
	GIN	1.165	175.82	90.66	110.74	0.539	0.691	848.24	1090.36	35.110	1498.23	1364.18	108.331
	D-MPNN	0.568	118.42	85.01	86.20	0.441	0.241	423.14	458.39	24.816	470.01	445.91	29.291
	PNA	0.681	148.88	88.72	97.31	0.361	0.409	664.98	692.74	23.855	616.70	694.92	57.217
	Graphormer	2.836	79.27	54.24	52.42	0.330	0.080	2066.28	2546.01	131.158	2229.88	2525.51	144.595
	AWARE	0.297	144.91	133.89	98.86	0.602	0.129	86.62	94.47	22.180	93.59	95.73	5.275
	GraphGPS	0.209	75.98	54.75	54.53	0.288	0.089	528.50	693.19	12.488	296.00	411.16	49.888
3D Graph	SchNet	0.060	44.13	27.64	22.55	0.028	0.031	14.19	14.05	0.133	13.93	13.27	1.749
	DimeNet++	0.044	36.22	20.01	16.66	0.028	0.022	7.45	6.14	0.323	6.33	7.18	1.118
	SE(3)-Trans	0.137	56.52	34.65	34.41	0.050	0.063	65.28	70.70	1.747	68.92	68.88	5.428
	EGNN	0.062	49.56	30.08	24.98	0.029	0.030	10.01	9.14	0.089	9.28	9.08	1.519
	PaiNN	0.049	42.73	24.46	20.16	0.016	0.025	8.43	7.88	0.169	8.18	7.63	1.419
	GemNet-T	0.041	35.46	17.85	15.86	0.021	0.023	7.61	7.08	0.271	6.42	5.88	1.232
	SphereNet	0.047	38.93	21.45	18.25	0.027	0.025	8.16	13.68	0.288	6.77	7.43	1.295
	SEGNN	0.048	33.61	17.66	17.01	0.021	0.026	11.60	12.45	0.404	11.29	12.20	1.590
	Allegro	0.097	102.44	61.86	63.17	0.176	0.032	42.08	44.96	1.977	44.64	44.43	2.949
	NequIP	0.066	61.94	42.00	31.64	0.036	0.028	22.08	23.36	0.415	23.23	23.02	1.899
	Equiformer	0.051	33.46	17.93	16.85	0.015	0.023	14.49	14.60	0.433	14.88	13.78	2.342
	Graphormer-3D	0.063	48.58	32.58	26.80	0.042	0.031	16.11	15.79	0.258	16.85	16.21	1.788
3D Transformer (w/ tokenization)	Geo2Seq	1.061	514.95	271.77	467.47	0.774	1.080	598.63	601.19	72.554	600.70	607.10	25.606
	InertialTransformer	0.074	45.32	27.60	25.92	0.046	0.034	23.23	25.97	0.723	25.74	25.19	2.050

Table 1: Results on 12 quantum mechanics prediction tasks in QM9, with 110K for training, 10K for validation, and 11K for testing. The task unit is specified, and the evaluation is the mean absolute error (MAE).

$[\text{RBF}(i,0), \text{RBF}(i,1), ..., \text{RBF}(i,m)]^T \in \mathbb{R}^{m\times 1}$. The approximated RBF(i,j) can be obtained as:

$$\begin{aligned} \tilde{k}(i,j) &= k_i^T A^{-1} k_j \\ &= \left(A^{-1/2} k_i\right)^T \left(A^{-1/2} k_j\right) \\ &= \left(L^{-1} k_i\right)^T \left(L^{-1} k_j\right), \end{aligned} \tag{8}$$

where $A = LL^T$ is the Cholesky decomposition.

Discussion. There is another research line using random features (*e.g.*, random Fourier features) for the pairwise distance approximation (Rahimi & Recht, 2007). There are certain works that have proved that Nyström method is more accurate (Yang et al., 2012). One intuitive way to understand this is that Nyström method utilizes the data-dependent basis, while the random features use data-independent basis functions.

4 Results

We use QM9 dataset (Ramakrishnan et al., 2014) to evaluate the performance of InertialTransformer for the prediction of scalar properties across chemical compound space. QM9 is a dataset consisting of 134K molecules, each with up to 9 heavy atoms. It includes 12 tasks that are related to the quantum properties. For example, U0 and U are the internal energies at temperatures of 0K and 298.15K, respectively, and U and G are the other two energies that can be calculated from H. The other 8 tasks are quantum mechanics related to the density functional theory (DFT) process. The data partition we use has 110k, 10k, and 11k molecules in training, validation and testing sets. We minimize mean absolute error (MAE) between prediction and normalized ground truth.

Section 4 shows the mean absolute error (MAE) of InertialTransformer for 12 target properties of QM9 in comparison with previous approaches. Compared to 3D Transformer baseline that employs structure tokenization (Li et al., 2024), our model consistently outperforms across all tasks, achieving lower MAE values with notable improvements. However, when evaluated against 3D GNN baselines, while our model surpasses certain established 3D GNN models (Musaelian et al. (2022) & Fuchs et al. (2020)), it remains competitive but does not exceed the performance of some state-of-the-art 3D GNN models such as Schütt et al. (2021).

5 Conclusion

Future Directions. InertialTransformer has the potential to be adopted to solve more challenging problems, such as protein folding or protein structure prediction. These are important questions, thus we would like to leave them for future exploration. On the other hand, InertialTransformer enables the modeling of molecules to be naturally combined with the attention and Transformer modules, which can be more effectively and efficiently merged into the multi-modal paradigms.

References

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016): 493–500, 2024.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Luis M Antunes, Keith T Butler, and Ricardo Grau-Crespo. Crystal structure generation with autoregressive large language modeling. *Nature Communications*, 15(1):1–16, 2024.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Daniel Flam-Shepherd and Alán Aspuru-Guzik. Language models can generate molecules, materials, and protein binding sites directly in three dimensions as xyz, cif, and pdb files, 2023. URL https://arxiv.org/abs/2305.05708.

Cong Fu, Xiner Li, Blake Olson, Heng Ji, and Shuiwang Ji. Fragment and geometry aware tokenization of molecules for structure-based drug design using language models, 2024. URL https://arxiv.org/abs/2408.09730.

Fabian B. Fuchs, Daniel E. Worrall, Volker Fischer, and Max Welling. Se(3)-transformers: 3d roto-translation equivariant attention networks, 2020. URL https://arxiv.org/abs/2006.10503.

Johannes Gasteiger, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs, 2022. URL https://arxiv.org/abs/2003.03123.

Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 54(10s):1–41, 2022.

Xiner Li, Limei Wang, Youzhi Luo, Carl Edwards, Shurui Gui, Yuchao Lin, Heng Ji, and Shuiwang Ji. Geometry informed tokenization of molecules for language model generation. *arXiv preprint arXiv:2408.10120*, 2024.

Yi-Lun Liao and Tess Smidt. Equiformer: Equivariant graph attention transformer for 3d atomistic graphs, 2023. URL https://arxiv.org/abs/2206.11990.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Shengchao Liu, Hanchen Wang, Weiyang Liu, Joan Lasenby, Hongyu Guo, and Jian Tang. Pre-training molecular graph representation with 3d geometry. *arXiv preprint arXiv:2110.07728*, 2021.

Shengchao Liu, Yanjing Li, Zhuoxinran Li, Zhiling Zheng, Chenru Duan, Zhi-Ming Ma, Omar Yaghi, Animashree Anandkumar, Christian Borgs, Jennifer Chayes, et al. Symmetry-informed geometric representation for molecules, proteins, and crystalline materials. *Advances in neural information processing systems*, 36:66084–66101, 2023.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey, 2025. URL https://arxiv.org/abs/2402.06196.

Albert Musaelian, Simon Batzner, Anders Johansson, Lixin Sun, Cameron J. Owen, Mordechai Kornbluth, and Boris Kozinsky. Learning local equivariant representations for large-scale atomistic dynamics, 2022. URL https://arxiv.org/abs/2204.05249.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20, 2007.

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.

Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E(n) equivariant graph neural networks, 2022. URL https://arxiv.org/abs/2102.09844.

Kristof Schütt, Oliver Unke, and Michael Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In *International Conference on Machine Learning*, pp. 9377–9388. PMLR, 2021.

Kristof T. Schütt, Pieter-Jan Kindermans, Huziel E. Sauceda, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions, 2017. URL https://arxiv.org/abs/1706.08566.

Yu Shi, Shuxin Zheng, Guolin Ke, Yifei Shen, Jiacheng You, Jiyan He, Shengjie Luo, Chang Liu, Di He, and Tie-Yan Liu. Benchmarking graphormer on large-scale molecular modeling datasets, 2023. URL https://arxiv.org/abs/2203.04810.

Jianlin Su. Road to transformer upgrades: 4. rotational positional encoding for 2d positions, May 2021. URL https://spaces.ac.cn/archives/8397.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971.

A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Christopher Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. *Advances in neural information processing systems*, 13, 2000.

Keqiang Yan, Xiner Li, Hongyi Ling, Kenna Ashen, Carl Edwards, Raymundo Arróyave, Marinka Zitnik, Heng Ji, Xiaofeng Qian, Xiaoning Qian, et al. Invariant tokenization of crystalline materials for language model enabled generation. *Advances in Neural Information Processing Systems*, 37:125050–125072, 2024.

Tianbao Yang, Yu-Feng Li, Mehrdad Mahdavi, Rong Jin, and Zhi-Hua Zhou. Nyström method vs random fourier features: A theoretical and empirical comparison. *Advances in neural information processing systems*, 25, 2012.