Revisiting the Form of Attention with Positional Encoding for Molecular Structures

Yusei Ito [1 2] **Aidar Alimbayev** [* 1] **Klea Ziu** [* 1] **Deepak Kumar** [* 1] **Kanta Ono** [2] **Martin Takáč** [1]

Abstract

Molecular property prediction is a core task in machine learning for physics, with applications to interatomic potential learning and materials discovery. Recent transformer-based models achieve strong performance, but their gains often conflate two coupled choices: the design of SE(3)-aware positional encodings and how these encodings are incorporated into attention. Consequently, it remains unclear whether the empirical improvements stem from the positional encoding itself or from the attention formulation used to exploit it. Here, we present a controlled study of attention design in molecular transformers that fixes the positional encoding and only varies the attention formulation. Our results show that attention design alone has a significant effect on performance. Under a shared positional encoding, vector-valued positional encodings are most effective when used to modulate attention logits, while the best overall results are obtained when geometric information is introduced through a separate pathway that is adaptively gated by the main attention message. Our study isolates the effect of attention design, identifies key ingredients for accurate prediction of molecular properties, and offers practical guidelines for future research.

1. Introduction

Molecular property prediction aims to map molecular structures to physical properties such as total energy, dipole moment, and HOMO–LUMO gap (Xie & Grossman, 2018; Chen & Ong, 2022; Liao & Smidt, 2023; Aykent & Xia, 2025). This is a central problem in machine learning for physics, as it enables the construction of accurate surrogates for computationally expensive *ab initio* calculations (Hohenberg & Kohn, 1964; Kohn & Sham, 1965) and thereby supports a wide range of applications, including machine-learning interatomic potentials (Batatia et al., 2022; Wood et al., 2025; Jacobs et al., 2025), drug design (Catacutan et al., 2024), and materials discovery (Pollice et al., 2021).

A key challenge in molecular modeling is that molecules are naturally represented as variable-sized sets of atoms in three-dimensional space, with no canonical ordering of atoms or coordinate system. Predictive models, therefore, need to be invariant to atom ordering and global SE(3)-transformations (*i.e.*, translations and rotations). These considerations initially motivated graph-based molecular representations, with graph neural networks (GNNs) emerging as the dominant framework through SE(3)-invariant or equivariant message-passing mechanisms (Xie & Grossman, 2018; Gasteiger et al., 2020; Chen & Ong, 2022). Although this GNN paradigm has been highly successful, more recent work has increasingly turned to transformer-based architectures, whose self-attention mechanisms offer a flexible way to model molecular interactions beyond local message passing (Ying et al., 2021; Kwak et al., 2023; Aykent & Xia, 2025).

In contrast to GNNs, where architectural variation often centers on the design of the message-passing mechanism itself, transformers largely standardize the basic interaction pattern through attention-based aggregation (Vaswani et al., 2017). As a result, the main architectural questions shift away from how information is passed between atoms and toward the treatment of molecular geometry. In this setting, two design choices are especially important: the construction of positional encodings that capture molecular geometry while preserving the required SE(3)-symmetries, and the formulation of the attention mechanism, which determines how geometric information modulates attention computations.

Most prior work on molecular transformers has focused on designing SE(3)-aware positional encodings (Wang et al., 2023; Aykent & Xia, 2025). In many cases, however, the attention mechanism is also modified simultaneously, making it unclear how much of the observed performance differences can be attributed to the positional encoding itself versus the associated attention design. This leads to a central

[*]Equal contribution [1]Mohamed bin Zayed University of Artificial Intelligence [2]The University of Osaka. Correspondence to: Martin Takáč <martin.takac@mbzuai.ac.ae>.

Proceedings of the AI4Physics Workshop at the 43rd International Conference on Machine Learning (AI4Physics@ICML 2026), Seoul, South Korea. 2026. Copyright 2026 by the author(s).

question: *under a fixed positional encoding, what role does attention design play in molecular property prediction?*

In this work, we address this question through a controlled study of molecular transformers in which the positional encoding is held fixed and only the attention formulation varies. Using a shared geometric representation, we compare different ways of incorporating positional information into attention, including the injection point and gating mechanisms (Dey & Salem, 2017; Qiu et al., 2025). This setup isolates the effect of attention design, clarifies which components matter most for strong performance, and offers practical guidance for the future development of transformer-based models.

We summarize our contributions as follows: (1) A unified formulation that subsumes prior molecular transformer architectures as special cases. (2) A controlled study on the QM9 and Molecule3D benchmarks that isolates the effects of attention mechanisms under a fixed positional-encoding scheme. (3) An empirical comparison showing that the configuration with vector positional encodings inside the attention softmax and a value vector-gated external branch achieves the strongest performance. (4) A mechanistic analysis that explains the effectiveness of this best-performing configuration.

2. Preliminaries

We begin by formally defining molecular structure and the general transformer architecture for molecules in Secs. 2.1 and 2.2. We then introduce the positional encoding function and describe its scalar and vector instantiations in Sec. 2.3.

2.1. Molecular structures

A molecular structure is characterized by the types of its constituent atoms and their three-dimensional spatial arrangement. Let the atomic species (atomic number) be denoted by

$$\mathcal{Z} = \{z_1, z_2, \ldots, z_N\}, \quad z_i \in \mathbb{Z}, \tag{1}$$

and the corresponding atomic positions by

$$\mathcal{P} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_N\}, \quad \boldsymbol{p}_i \in \mathbb{R}^3. \tag{2}$$

The molecular structure is then formally defined as $\mathcal{M} = (\mathcal{Z}, \mathcal{P})$. In this work, we consider the task of predicting a scalar molecular property $y \in \mathbb{R}$ from the molecular structure $\mathcal{M}$.

An important requirement for a predictive model f is permutation invariance with respect to the ordering of atoms, since the indexing of atoms is arbitrary. In addition, a model f should be invariant under SE(3)-transformations (*i.e.*, rotations and translations), which preserve the physical identity of the molecular structure. Let $Q \in \mathrm{SO}(3)$ be a rotation matrix and $\boldsymbol{t} \in \mathbb{R}^3$ be a translation vector. The SE(3)-invariance condition can be expressed as $f(\mathcal{Z}, \mathcal{P}) = f\big(\mathcal{Z}, \{Q\boldsymbol{p}_i + \boldsymbol{t}\}_{i=1}^N\big)$. This invariance can be enforced by constraining the model to use only SE(3)-invariant positional information, such as interatomic distances.

2.2. Transformer for molecular structures

In transformer-based models for molecular structures, each token corresponds to an individual atom in the molecule. Given a molecular structure $\mathcal{M} = (\mathcal{Z}, \mathcal{P})$, the initial atom-level representations are obtained through an embedding module:

$$\mathcal{H}^{(0)} = \mathrm{Emb}(\mathcal{M}) = \{\boldsymbol{h}_1^{(0)}, \ldots, \boldsymbol{h}_N^{(0)}\}, \quad \boldsymbol{h}_i^{(0)} \in \mathbb{R}^d. \tag{3}$$

Here, $\mathrm{Emb}(\cdot)$ denotes an embedding function that maps the molecular structure to atom-wise d-dimensional representations (*e.g.*, one-hot encodings of the atomic species).

The initial atom embeddings are then iteratively refined through L attention-based layers, allowing each atom to incorporate information from other atoms and their geometric relationships. Assuming the SE(3)-invariant geometric encoding $\mathcal{G}_{ij}$ between atom i and j, whose specific form will be discussed in the next section, the atom embeddings are updated according to the following general formula:

$$\tilde{\boldsymbol{h}}_i^{(\ell)} = \boldsymbol{h}_i^{(\ell)} + \underbrace{\sum_{j=1}^{N} \alpha_{ij}^{(\ell)} \boldsymbol{m}_{ij}^{(\ell)}}_{\text{Main pathway}} + \underbrace{\sum_{j=1}^{N} \beta_{ij}^{(\ell)} \boldsymbol{n}_{ij}^{(\ell)}}_{\text{External pathway}}, \tag{4}$$

followed by an atom-wise feed-forward network:

$$\boldsymbol{h}_i^{(\ell+1)} = \tilde{\boldsymbol{h}}_i^{(\ell)} + \mathrm{FFN}\Big(\tilde{\boldsymbol{h}}_i^{(\ell)}\Big). \tag{5}$$

Here, $\alpha_{ij}^{(\ell)}$ denotes the attention weight from atom j to atom i, and $\boldsymbol{m}_{ij}^{(\ell)}$ denotes the corresponding message in the main attention pathway. $\boldsymbol{n}_{ij}^{(\ell)}$ and $\beta_{ij}^{(\ell)}$ denote an additional message and its coefficient, which are defined separately from the main attention term, allowing them to convey supplementary geometric information in parallel with the main attention branch. Both pathways may depend on the atom embeddings and the pairwise geometric encoding $\mathcal{G}_{ij}$. After L layers, the final atom-wise representations $\{\boldsymbol{h}_i^{(L)}\}_{i=1}^N$ are aggregated through a permutation-invariant aggregation function (*e.g.*, mean pooling) to obtain a molecule-level representation, which is then fed into a feed-forward network to predict the target scalar property.

This general formulation, shown in Eq. (4), includes the standard attention formulation. When geometric information is omitted and the additional external pathway is removed, this formulation reduces to the standard self-attention mech-

anism by defining $\alpha_{ij}^{(\ell)}$ and $\boldsymbol{m}_{ij}^{(\ell)}$ as follows:

$$\alpha_{ij}^{(\ell)} = \mathrm{softmax}_j\left(\frac{(\boldsymbol{q}_i^{(\ell)})^\top \boldsymbol{k}_j^{(\ell)}}{\sqrt{d}}\right), \quad \boldsymbol{m}_{ij}^{(\ell)} = \boldsymbol{v}_j^{(\ell)}, \quad (6)$$

where $\boldsymbol{q}_i^{(\ell)} = W_Q \boldsymbol{h}_i^{(\ell)}$, $\boldsymbol{k}_j^{(\ell)} = W_K \boldsymbol{h}_j^{(\ell)}$, $\boldsymbol{v}_j^{(\ell)} = W_V \boldsymbol{h}_j^{(\ell)}$, and W_Q, W_K, and W_V are trainable matrices. Previous studies generally have extended the standard self-attention formulation in Eq. (6) by incorporating geometric encoding $\mathcal{G}_{ij}$ in various ways. Because both the form of $\mathcal{G}_{ij}$ and the method of incorporation differ across studies, it is difficult to disentangle the respective effects of the geometric encoding and the interaction formulation. In the next section, we explain that geometric encoding $\mathcal{G}_{ij}$ can be incorporated into the attention mechanism in two broad ways: scalar positional encodings and vector positional encodings.

2.3. Scalar and vector positional encodings

In transformer-based interaction formulations, where atom embeddings are represented as d-dimensional vectors, pairwise geometric encodings $\mathcal{G}_{ij}$ can be naturally divided into two types: scalar positional encodings, $\phi_{ij}^{(\ell)} \in \mathbb{R}$, and vector positional encodings, $\boldsymbol{\psi}_{ij}^{(\ell)} \in \mathbb{R}^d$. The former enter the formulation as scalar quantities, whereas the latter are introduced as d-dimensional vectors that act element-wise on the embeddings.

Scalar positional encodings $\phi_{ij}^{(\ell)}$ are introduced as scalar terms in message-passing calculations. A common example is distance-based encoding, such as a cosine or Gaussian decay as a function of the interatomic distance (Wang et al., 2023). These encodings can be incorporated into the attention computation in several ways. For example, given a scalar positional encoding $\phi_{ij}^{(\ell)}$, one may add it to the attention logit:

$$\alpha_{ij}^{(\ell)} = \mathrm{softmax}_j\left(\frac{(\boldsymbol{q}_i^{(\ell)})^\top \boldsymbol{k}_j^{(\ell)}}{\sqrt{d}} + \phi_{ij}^{(\ell)}\right). \quad (7)$$

Alternatively, the scalar encoding may be used to modulate the message multiplicatively: $\boldsymbol{m}_{ij}^{(\ell)} = \phi_{ij}^{(\ell)} \boldsymbol{v}_j^{(\ell)}$. In both cases, the scalar positional encoding essentially modulates the interaction strength, typically attenuating it as the interatomic distance increases.

By contrast, vector positional encodings lie in the same dimensional space as the atom-wise representations and can therefore interact with each embedding in an element-wise manner. The simplest approach is to map the scalar interatomic distance to a vector using a learnable function (Schütt et al., 2017), while recent studies have also proposed vector positional encodings that incorporate angular information (Wang et al., 2023; Aykent & Xia, 2025). Let $\boldsymbol{\psi}_{ij}^{(\ell)} \in \mathbb{R}^d$ denote a vector positional encoding for the atom pair (i, j). A straightforward approach is to use $\boldsymbol{\psi}_{ij}^{(\ell)}$ to modulate the transmitted message as $\boldsymbol{m}_{ij}^{(\ell)} = \boldsymbol{v}_j^{(\ell)} + \boldsymbol{\psi}_{ij}^{(\ell)}$, or to apply it to the attention logit as

$$\alpha_{ij}^{(\ell)} = \mathrm{softmax}_j\left(\frac{(\boldsymbol{q}_i^{(\ell)})^\top (\boldsymbol{k}_j^{(\ell)} \odot \boldsymbol{\psi}_{ij}^{(\ell)})}{\sqrt{d}}\right). \quad (8)$$

where $\odot$ denotes element-wise multiplication. Unlike scalar positional encodings, which primarily act as distance-dependent modifiers of interaction strength, vector positional encodings can directly reshape vector-valued representations, thereby enabling a more flexible encoding of geometric information. A detailed description of the forms of positional encodings adopted in prior studies is provided in Appendix A.

As illustrated by these examples, both scalar and vector positional encoding functions can be incorporated in various ways. In this work, we systematically compare these formulations to determine which are most effective when using the fixed positional encoding functions $\phi_{ij}^{(\ell)}$ and $\boldsymbol{\psi}_{ij}^{(\ell)}$.

3. Formulation design space

Our goal is to compare different attention formulations while holding fixed the functional forms of the scalar and vector positional encodings. To this end, as shown in Fig. 1, we first define the formulation design space comprising three locations where positional information can be injected into the standard attention formulation in Eq. (6): (i) ***inside*** the softmax, where positional information modifies the attention logits; (ii) ***outside*** the softmax, where positional information modulates the transmitted message; and (iii) an ***external*** pathway, where a pairwise geometric term contributes an additional message in parallel with the main attention branch.

Let the scalar terms associated with these three locations be $(\phi_{ij}^{\mathrm{in}}, \phi_{ij}^{\mathrm{out}}, \phi_{ij}^{\mathrm{ext}})$, and the corresponding vector-valued terms be $(\boldsymbol{\psi}_{ij}^{\mathrm{in}}, \boldsymbol{\psi}_{ij}^{\mathrm{out}}, \boldsymbol{\psi}_{ij}^{\mathrm{ext}})$. For simplicity, we omit the layer index ℓ. A unified template for the formulations considered in this work is given by

$$\begin{aligned} \alpha_{ij}^{(\ell)} &= \mathrm{softmax}_j\left(\frac{(\boldsymbol{q}_i^{(\ell)})^\top \left(\boldsymbol{k}_j^{(\ell)} \odot \boldsymbol{\psi}_{ij}^{\mathrm{in}}\right)}{\sqrt{d}} + \phi_{ij}^{\mathrm{in}}\right), \\ \boldsymbol{m}_{ij}^{(\ell)} &= \phi_{ij}^{\mathrm{out}} \cdot \left(\boldsymbol{v}_j^{(\ell)} + \boldsymbol{\psi}_{ij}^{\mathrm{out}}\right), \\ \beta_{ij}^{(\ell)} &= \phi_{ij}^{\mathrm{ext}}, \quad \boldsymbol{n}_{ij}^{(\ell)} = \boldsymbol{\psi}_{ij}^{\mathrm{ext}}. \end{aligned} \quad (9)$$

The three injection locations are expected to play distinct roles. The inside-softmax terms, ϕ_{ij}^{in} and $\boldsymbol{\psi}_{ij}^{\mathrm{in}}$, modify the attention logits and therefore control how strongly atom i attends to atom j. By contrast, the outside-softmax terms,

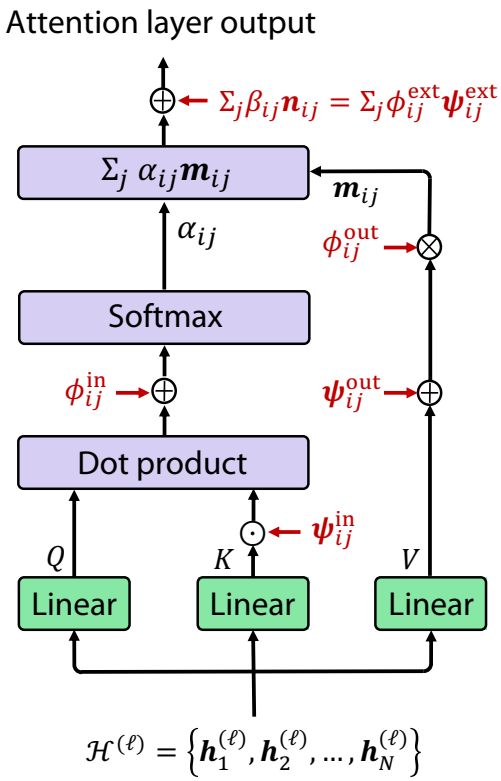

Figure 1. Schematic illustration of the formulation design space considered in this work. Given atom-wise representations $\mathcal{H}^{(\ell)}$, scalar and vector positional encodings, ϕ_{ij} and $\boldsymbol{\psi}_{ij}$, can be injected into the attention block in three ways: inside the softmax, outside the softmax, or through an external pathway.

ϕ_{ij}^{out} and $\boldsymbol{\psi}_{ij}^{\text{out}}$, do not affect interaction strength; instead, they modify the message passed through each atom pair. Finally, the external pathway $\beta_{ij}^{(\ell)} \boldsymbol{n}_{ij}^{(\ell)} = \phi_{ij}^{\text{ext}} \boldsymbol{\psi}_{ij}^{\text{ext}}$ provides an additional pairwise interaction parallel to the main attention branch, with potentially completely different interactions. *Each term is motivated by a plausible modeling intuition, making it an intriguing question which point of geometric injection offers the greatest benefit.*

In addition to the injection location, we investigate whether vector-valued positional terms benefit from gating (Dey & Salem, 2017; Qiu et al., 2025), which has also been used in previous methods (Aykent & Xia, 2025). The gating mechanism is defined as $\hat{\boldsymbol{\psi}}_{ij} = \boldsymbol{g}_{ij} \odot \boldsymbol{\psi}_{ij}$, where $\boldsymbol{g}_{ij} = \text{Act}(W\boldsymbol{x})$, $\text{Act}(\cdot)$ is an activation function (*e.g.*, SiLU, Sigmoid function), $\boldsymbol{x} \in \mathbb{R}^d$ denotes the atom-wise or atom-pair-wise feature embedding, and $W \in \mathbb{R}^{d \times d}$ is a trainable matrix. The intuition is that geometric information is not equally informative across all atom pairs or feature channels; its relevance may vary depending on the atoms involved.

Overall, the design space considered in this work is defined by two factors: (i) the injection location of the scalar and vector positional terms, and (ii) whether gating is used, together with the choice of input feature embeddings $\boldsymbol{x}$. By varying these factors while keeping the positional encoding fixed, we isolate the effect of the attention formulation itself. As we discuss in Appendix A, this design space is broad enough to subsume the formulations used in prior work. [1]

4. Experiments

4.1. Experimental setups

To compare the attention formulations introduced in Sec. 3 under fair conditions, we fix all other aspects of the experimental setup, including the overall architecture, the positional encoding functions (see next paragraph), the training parameters, and the evaluation benchmark; details of the architecture and training setup are provided in Appendix B. Specifically, we use an 8-layer, 8-head transformer with 256-dimensional atom representations, and the model has 9.74M parameters when no positional encoding is used.

Positional encoding functions. For the scalar positional encoding, we adopt a cosine cutoff function with a cutoff radius of $c = 5$ Å (Wang et al., 2023). Let $d_{ij} = \|\boldsymbol{p}_j - \boldsymbol{p}_i\|_2$ denote the distance between atoms i and j. We define the scalar encoding as $\exp(\phi_{ij}^{\text{in}}) = \phi_{ij}^{\text{out}} = \phi_{ij}^{\text{ext}} = \cos(\pi d_{ij}/2c)$, for $d_{ij} < c$, and as 0 otherwise.

For the vector positional encoding, we employ a Gaussian radial basis expansion of the interatomic distance (Schütt et al., 2017). Specifically, we first construct a K-dimensional basis vector $\boldsymbol{b}_{ij} = [b_1(d_{ij}), b_2(d_{ij}), \ldots, b_K(d_{ij})]$, where each basis function is defined as

$$b_k(d_{ij}) = \exp\left(-\frac{(d_{ij} - \mu_k)^2}{2(r_{\max}/K)^2}\right), \quad \mu_k = \frac{k \cdot r_{\max}}{K}. \quad (10)$$

Here, $r_{\max}$ and K are hyperparameters, which we set to 10 Å and 64, respectively. The final vector-valued positional encoding is obtained through a learned linear projection: $\psi_{ij} = W\boldsymbol{b}_{ij}$, where $W \in \mathbb{R}^{d \times K}$ is a trainable matrix. This encoding can be interpreted as a distance-based soft one-hot representation spanning from 0 Å to $r_{\max}$ projected into the model's feature space.

The number of parameters for each method is as follows: scalar encoding has no trainable parameters, whereas vector encoding introduces a 64×256 matrix over 8 layers, for a total of 0.13M parameters. In addition, when gating is used, an extra 256×256 matrix is required for each of the 8 layers, adding 0.52M parameters.

[1] Several methods replace the softmax function with alternatives such as SiLU. However, in this paper, we use softmax throughout to focus on the attention formulation.

Table 1. Results of a controlled study on scalar and vector positional encodings using the QM9 dataset, where the vector positional encoding is varied first, followed by the scalar positional encoding. Check marks indicate the positional encodings used. In each subsection, the best and second-best results are highlighted in bold and with underlining, respectively.

#	ϕ^{in}	ϕ^{out}	$\boldsymbol{\psi}^{\text{in}}$	$\boldsymbol{\psi}^{\text{out}}$	Added param (Million)	α ($\text{m}a_0^3$)	$\Delta\epsilon$ (meV)	μ (mD)	U_0 (meV)	ZPVE (meV)	std. (%)	log -
0					0 (+0.0%)	71.31	54.60	116.9	12.62	1.617	2.635	-3.699
(I) Comparison of vector positional encodings												
1	✓				0 (+0.0%)	67.53	48.26	47.04	12.53	1.563	1.599	-3.925
2	✓		✓		0.13 (+1.3%)	58.73	**39.76**	35.59	10.63	1.472	1.287	-4.092
3	✓			✓	0.13 (+1.3%)	65.02	46.55	43.21	12.19	1.569	1.515	-3.961
4	✓		✓	✓	0.26 (+2.7%)	**55.51**	39.92	**34.79**	**10.60**	**1.415**	**1.270**	**-4.116**
(II) Comparison of scalar positional encodings												
(i) Fixed vector encoding: $\boldsymbol{\psi}^{in}$												
5			✓		0.13 (+1.3%)	**55.40**	43.33	31.06	11.06	1.480	1.276	-4.105
2	✓		✓		0.13 (+1.3%)	58.73	**39.76**	35.59	**10.63**	1.472	1.287	-4.092
6		✓	✓		0.13 (+1.3%)	57.99	42.85	**30.09**	11.22	1.441	**1.261**	**-4.107**
7	✓	✓	✓		0.13 (+1.3%)	57.97	42.02	40.50	11.04	**1.437**	1.386	-4.055
(ii) Fixed vector encoding: $\boldsymbol{\psi}^{in}, \boldsymbol{\psi}^{out}$												
8			✓	✓	0.26 (+2.7%)	**55.48**	44.30	35.85	**10.42**	1.457	1.352	-4.086
4	✓		✓	✓	0.26 (+2.7%)	55.51	**39.92**	34.79	10.60	**1.415**	**1.270**	**-4.116**
9		✓	✓	✓	0.26 (+2.7%)	60.37	42.48	**31.04**	10.91	1.464	1.274	-4.097
10	✓	✓	✓	✓	0.26 (+2.7%)	60.06	42.50	38.77	11.04	1.451	1.376	-4.053

Dataset and evaluation metrics. We evaluate all methods on five molecular property prediction tasks from QM9, a common benchmark for molecular modeling (Ramakrishnan et al., 2014). These targets form a compact yet physically diverse subset of QM9, covering electronic response (α), frontier-orbital electronic structure ($\Delta\epsilon$), electrostatics (μ), thermochemical energetics (U_0), and vibrational quantum effects (ZPVE). The dataset contains 130,831 molecules, which we split into 110,000 training samples, 10,000 validation samples, and the remainder for testing, following prior work (Aykent & Xia, 2025). Mean absolute error (MAE) is used as the primary evaluation metric for each task. To summarize performance across tasks with different target scales, we also report two scale-invariant aggregate metrics. The first is a normalized error, denoted by ***std.***, which is computed by dividing the MAE for each task by its target values' standard deviation and then averaging the resulting normalized errors across all tasks. The second is a logarithmic aggregate metric, denoted by ***log***, which is computed by taking the logarithm of the MAE for each task and then averaging these values across tasks. Together, these two metrics complement the per-task MAE by providing cross-task summaries that are less sensitive to differences in target magnitude. We formally define these two metrics in Appendix C.

4.2. Inside vs. outside positional encoding

We begin by considering only the calculation of the main attention in Eq. (4), ignoring the third term corresponding to the external interaction pathway. Table 1 (I) (settings (1)–(4)) shows the results obtained by fixing the scalar positional encoding to the *inside* formulation while varying how the vector-valued positional encoding is used. Specifically, we compare: (1) a model without any vector-valued positional encoding; (2)–(3) models that use the vector positional encoding either *inside* or *outside* the softmax function; and (4) a model that uses it in both (*inside* + *outside*) locations. For reference, setting (0) presents a baseline model with no positional encoding at all.

The results show a clear trend: *using the vector positional encoding inside the softmax function performs better than using it only outside the softmax.* The inside-softmax configuration outperformed the outside-softmax configuration across all five tasks, achieving a 15% reduction in normalized error. We discuss this observation further in Sec. 5.1. Additionally, using the vector positional encoding both inside + outside the softmax function yields slightly better performance than using it only inside, but the improvement is modest.

Next, we compare different ways of incorporating scalar

Table 2. Comparison of outside and external vector positional encodings, with and without SiLU gating, as well as the feature embeddings used for gating, on the QM9 dataset. "Gate" denotes the feature used to compute the gate. In each subsection, the best and second-best results are highlighted in bold and with underlining, respectively.

#	ϕ^{in}	ϕ^{out}	ϕ^{ext}	ψ^{in}	ψ^{out}	ψ^{ext}	Gate	Added param (Million)	α ($\text{m}a_0^3$)	$\Delta\epsilon$ (meV)	μ (mD)	U_0 (meV)	ZPVE (meV)	std. (%)	log -
(I) Comparison of gating for outside vector positional encoding															
4	✓			✓	✓		–	0.26 (+2.7%)	55.51	39.92	34.79	10.60	1.415	1.270	-4.116
11	✓			✓	✓		$\boldsymbol{v}_j$	0.79 (+8.1%)	**53.80**	**39.23**	**34.02**	11.33	1.429	**1.247**	-4.115
12	✓			✓	✓		$\boldsymbol{h}_j$	0.79 (+8.1%)	54.14	40.11	35.22	10.86	1.433	1.276	-4.110
13	✓			✓	✓		$\boldsymbol{h}_i + \boldsymbol{h}_j$	0.79 (+8.1%)	55.16	40.58	34.37	**10.09**	**1.364**	1.272	**-4.133**
(II) Comparison of gating for external vector positional encoding															
14	✓		✓	✓		✓	–	0.26 (+2.7%)	55.69	40.17	36.08	10.86	1.400	1.292	-4.104
15	✓		✓	✓		✓	$\boldsymbol{v}_j$	0.79 (+8.1%)	51.50	**36.85**	**32.88**	9.905	1.336	**1.184**	-4.182
16	✓		✓	✓		✓	$\boldsymbol{h}_j$	0.79 (+8.1%)	**50.79**	37.28	32.99	**9.505**	1.357	1.190	**-4.188**
17	✓		✓	✓		✓	$\boldsymbol{h}_i + \boldsymbol{h}_j$	0.79 (+8.1%)	54.17	38.63	33.53	9.773	**1.302**	1.226	-4.167
(III) Extended comparison with fixed gate $\boldsymbol{v}_j$															
15	✓		✓	✓		✓	$\boldsymbol{v}_j$	0.79 (+8.1%)	51.50	**36.85**	32.88	**9.905**	**1.336**	1.184	-4.182
18		✓	✓	✓		✓	$\boldsymbol{v}_j$	0.79 (+8.1%)	**51.31**	37.70	**27.34**	10.67	1.373	**1.126**	**-4.196**
19	✓		✓	✓	✓	✓	$\boldsymbol{v}_j$	1.44 (+14.8%)	51.83	37.17	31.55	10.11	1.340	1.173	-4.184

positional encoding, focusing on the two better-performing variants of vector positional encoding: the *inside* and *inside + outside* settings. The results are summarized in Table 1 (II): (i) settings (2), (5)–(7) use *inside*-vector positional encoding, while (ii) settings (4), (8)–(10) use both *inside*- and *outside*-vector positional encodings. Unlike the comparison of vector positional encoding functions, *scalar positional encoding does not exhibit a consistent trend across tasks.* One possible explanation is that placing scalar positional encoding inside or outside the softmax plays a similar role in both cases, since both formulations effectively attenuate interaction strength as a function of interatomic distance.

4.3. External positional encoding and gating

We next examine the effect of the third term in Eq. (4), which represents the external pathway, as well as the role of gating. Based on the findings in the previous section, we use the better-performing *inside + outside* vector encoding as the reference and compare it against variants in which the outside encoding is replaced by an external interaction pathway: *inside + external* scalar encoding and *inside + external* vector encoding. For both the *outside* vector encoding and the *external* vector encoding, we further evaluate formulations with and without gating, while also varying the feature embedding used. The results are summarized in Table 2: (I) settings (4) and (11)–(13) correspond to the outside formulation and its gating variants, while (II) settings (14)–(17) correspond to the external formulation and its gating variants.

For gating, we use the SiLU activation function, following prior work (Aykent & Xia, 2025), and compare three choices of input features for constructing the gate: settings (11, 15) use only the value vector $\boldsymbol{v}_j$; settings (12, 16) use the target atom-wise embedding $\boldsymbol{h}_j$; and settings (13, 17) use the atom-pair-wise embedding $\boldsymbol{h}_i + \boldsymbol{h}_j$. These comparisons allow us to examine not only whether gating is beneficial, but also which information source is most effective for modulating the vector positional encoding term.

Although both gated variants introduce the same increase in parameter count, the benefit of gating differs markedly between the two formulations. In the inside + outside setting, gating provides only marginal gains, improving performance on 3 out of 5 tasks and reducing normalized error by 1.8% in value vector case, whereas in the inside + external formulation, it consistently improves performance across all 5 tasks and reduces normalized error by 8.4%. Interestingly, replacing the outside formulation with the external one alone offers almost no benefit and may even slightly degrade performance in some cases. These findings show that *the improvement arises not from the external pathway itself, but from its combination with an appropriate gating mechanism.*

Among the gating variants, *the best performance is achieved when the gate is constructed from either the value vector $\boldsymbol{v}_j$ or the target's atom-wise embedding $\boldsymbol{h}_j$, whereas variants based on pair-wise information $\boldsymbol{h}_i + \boldsymbol{h}_j$ are less effective.* This implies that it is beneficial to modulate the external message based on the content of the main term message itself. We will further discuss the performance gain in Sec. 5.2.

Finally, Table 2 (III) compares the best-performing *inside + external* configurations for scalar and vector positional encodings with two variants: setting (18), which replaces the scalar encoding with an *outside* formulation, and setting (19), which further introduces *outside* vector positional en-

Table 3. Ablation study on two factors hypothesized to explain the advantage of inside-softmax vector positional encoding. Starting from the outside formulation, we test adding species information, replacing additive message injection with multiplicative modulation, and combining both modifications. Additionally, we report the results obtained by changing the inside formulation to the additive form. The best and second-best results are highlighted in bold and underlined, respectively.

#	ϕ^{in}	ϕ^{out}	ψ^{in}	ψ^{out}	ψ^{out} variant	Added param (Million)	α (ma_0^3)	$\Delta\epsilon$ (meV)	μ (mD)	U_0 (meV)	ZPVE (meV)	std. (%)	log -
Outside vector positional encoding variants													
3	✓			✓	base	0.13 (+1.3%)	65.02	46.55	43.21	12.19	1.569	1.515	-3.961
20	✓			✓	+ species info.	0.66 (+6.8%)	59.90	43.86	40.82	12.27	1.499	1.427	-4.009
21	✓			✓	multiplicative	0.13 (+1.3%)	<u>55.81</u>	43.53	50.53	11.57	<u>1.474</u>	1.539	-3.997
22	✓			✓	+ both	0.66 (+6.8%)	**51.01**	<u>42.14</u>	46.67	<u>10.87</u>	1.478	1.453	<u>-4.049</u>
Inside vector positional encoding													
2	✓		✓		base	0.13 (+1.3%)	58.73	**39.76**	**35.59**	**10.63**	**1.472**	**1.287**	**-4.092**
23	✓		✓		additive	0.13 (+1.3%)	64.50	45.00	<u>35.92</u>	11.92	1.553	<u>1.392</u>	-4.013

coding. Consistent with the results in Sec. 4.2, changing the scalar encoding remained task-dependent, and no consistent trend was observed across tasks. Moreover, adding outside vector positional encoding did not improve performance and even degraded it on some tasks. One possible reason is that, as discussed in Sec. 5.2, external encoding functions as a complement to the main term. Therefore, once positional encoding is also added to the main message, the separation of signals between the main term and the external term may become less clear.

In Appendices D, E, and F, we show that the findings also hold for a different positional encoding function, a different initialization of atom embeddings, and a different dataset.

Architecture design recipe

(i) For vector positional encoding, the inside-softmax formulation outperforms the outside-softmax formulation; and (ii) the external term is most effective when combined with value-vector gating.

5. Analysis

5.1. Why does inside-softmax injection work better?

As shown in Sec. 4.2, placing the vector positional encoding inside the softmax via $\boldsymbol{\psi}_{ij}^{\text{in}}$ (setting (2)) yields better performance than introducing it outside the softmax via $\boldsymbol{\psi}_{ij}^{\text{out}}$ (setting (3)), despite using the same positional representation and the same number of additional parameters. A comparison of the two formulations suggests that the advantage of inside-softmax injection may arise not only from injecting positional information into attention logits, but also from two structural differences: it naturally introduces atom-dependent co-occurrence information and incorporates positional information through multiplicative modulation.

First, the inside formulation includes the term $(\boldsymbol{q}_i^{(\ell)})^\top(\boldsymbol{k}_j^{(\ell)} \odot \boldsymbol{\psi}_{ij}^{\text{in}})$, which captures co-occurrence patterns between atom-dependent features and interatomic distances, thereby enabling the model to represent distance-conditioned interactions between atomic species. Second, the inside formulation incorporates positional information multiplicatively into the attention computation, whereas the outside formulation incorporates it additively. Compared with simple addition, multiplicative interaction may provide stronger modulation of the original representation.

To examine these two factors, we performed two additional experiments based on the lower-performing outside formulation (setting (3)). To test the role of atom-dependent co-occurrence information, we augmented the outside positional term as $\hat{\boldsymbol{\psi}}_{ij}^{\text{out}} = W(\boldsymbol{q}_i^{(\ell)} \odot \boldsymbol{k}_j^{(\ell)}) \odot \boldsymbol{\psi}_{ij}^{\text{out}}$, where $W \in \mathbb{R}^{d\times d}$ is a trainable matrix (setting (20)). To test the role of multiplicative interaction, we replaced the additive outside message $\boldsymbol{m}_{ij}^{(\ell)} = \boldsymbol{v}_j^{(\ell)} + \boldsymbol{\psi}_{ij}^{\text{out}}$ with the multiplicative form $\boldsymbol{m}_{ij}^{(\ell)} = \boldsymbol{v}_j^{(\ell)} \odot \boldsymbol{\psi}_{ij}^{\text{out}}$ (setting (21)). Conversely, starting from the inside formulation, we ablated its multiplicative interaction by replacing the multiplicative inside vector positional encoding with an additive form, namely $(\boldsymbol{q}_i^{(\ell)})^\top(\boldsymbol{k}_j^{(\ell)} + \boldsymbol{\psi}_{ij}^{\text{in}})$ (setting (23)).

As shown in Table 3, the variants in settings (20) and (21), as well as the combined variant in setting (22), outperform the original outside formulation (setting (3)) on four of the five tasks. These results indicate that both atom-dependent co-occurrence information and multiplicative modulation contribute meaningfully to the superiority of the inside formulation. The effectiveness of the multiplicative formulation is also supported by the fact that replacing the inside formulation with the additive one degraded performance across all five metrics as shown in setting (23).

We further analyzed the difference between additive and multiplicative outside injection by measuring the cosine similarity between the original value vector $\boldsymbol{v}_j^{(\ell)}$ and the

resulting message $\boldsymbol{m}_{ij}^{(\ell)}$: $\boldsymbol{v}_j^{(\ell)} \cdot \boldsymbol{m}_{ij}^{(\ell)} / (\|\boldsymbol{v}_j^{(\ell)}\| \, \|\boldsymbol{m}_{ij}^{(\ell)}\|)$. Figure 2 shows histograms constructed by pooling 1,000 randomly sampled test instances from each task across all layers and heads. The additive outside formulation produces a distribution sharply concentrated near 1.0, whereas the multiplicative formulation yields a much broader distribution whose peak is around 0.0. This suggests that additive injection often preserves the direction of the original message, whereas multiplicative modulation has a stronger influence on the interaction and can induce a different representation.

However, these two factors do not fully account for the advantage of the inside formulation shown in setting (2), which implies that the benefit of inside-softmax injection is not explained entirely by representational effects alone. A plausible interpretation is that directly injecting positional information into the attention logits provides an additional benefit. In the inside formulation, geometric information helps determine which atoms should interact, while in the outside formulation it can only modulate the message once the interaction weights have been fixed.

Our results therefore indicate that the advantage of inside-softmax injection is best understood as a combination of two effects: it naturally realizes a more expressive interaction form, through atom-dependent co-occurrence and multiplicative modulation, and it may further benefit from allowing positional information to directly influence attention routing.

5.2. Why does gated external injection work better?

As shown in Sec. 4.3, the external formulation with gating achieves the best performance among the configurations considered here. To better understand this gain, we analyze the fraction of the total message attributable to the positional-encoding-derived term. For the outside formulation, we measure $|\alpha_{ij}^{(\ell)} \boldsymbol{\psi}_{ij}^{\mathrm{out}}| / |\alpha_{ij}^{(\ell)} \boldsymbol{m}_{ij}^{(\ell)}|$, whereas for the external formulation, we measure $|\beta_{ij}^{(\ell)} \boldsymbol{n}_{ij}^{(\ell)}| / |\alpha_{ij}^{(\ell)} \boldsymbol{m}_{ij}^{(\ell)} + \beta_{ij}^{(\ell)} \boldsymbol{n}_{ij}^{(\ell)}|$, with all operations applied element-wise.

Figure 3 shows the resulting histograms for (a) *outside* and *outside + gating (value vector)*, and (b) *external* and *external + gating (value vector)*. As in Sec. 5.1, we randomly selected 1,000 instances from the test set for each task and used all layers and heads. Compared with outside, the external formulation exhibits a clear concentration near 1.0, indicating that the positional term can account for most of the total message for a nontrivial subset of elements. This behavior is consistent with the structure of the two formulations. In the outside case, the positional term is injected within the message of the main branch and shares the same attention coefficient α_{ij} as the value vector term, thus it primarily corrects the main message. In contrast, the external formulation introduces the positional term through a separate branch, allowing it to influence the message more independently and potentially alter which atom interacts most strongly. We discuss this further in Appendix G.

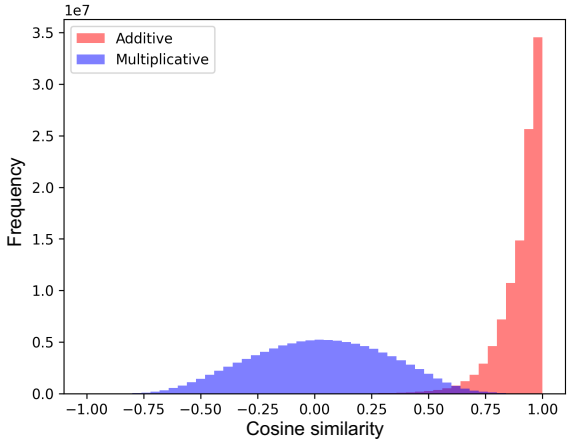

Figure 2. Distribution of cosine similarity between the original value vector $\boldsymbol{v}_j^{(\ell)}$ and the resulting message $\boldsymbol{m}_{ij}^{(\ell)}$ for additive and multiplicative injection methods. The additive formulation produces messages that remain highly aligned with the original value vector, whereas the multiplicative formulation yields a much broader distribution, indicating stronger interactions.

Gating affects the two formulations differently. In the outside case, gating shifts the distribution further toward 0.0, suggesting that the positional term is more often suppressed and used selectively as a correction to the main branch. In the external case, however, the histogram for external + gating is concentrated near both 0.0 and 1.0, yielding a bimodal pattern, which indicates that the external term is either strongly suppressed or becomes the dominant component of the total message. Such behavior implies that gating does not merely rescale the external branch uniformly; rather, it enables main message-dependent control over when and through which channels geometric information should be injected strongly.

In summary, combining the external term with a gate enables the model to capture information that the main term alone misses. The gate adaptively decides, for each dimension, whether the main or external term should dominate. Consequently, using pair-wise atom embeddings $\boldsymbol{h}_i + \boldsymbol{h}_j$, which are relatively distant from the main message, leads to inferior performance, whereas the best performance is achieved when using $\boldsymbol{v}_j$, *i.e.*, the main message itself.

6. Conclusion and future work

In this work, we revisited attention design in molecular transformers through a controlled study that decouples it

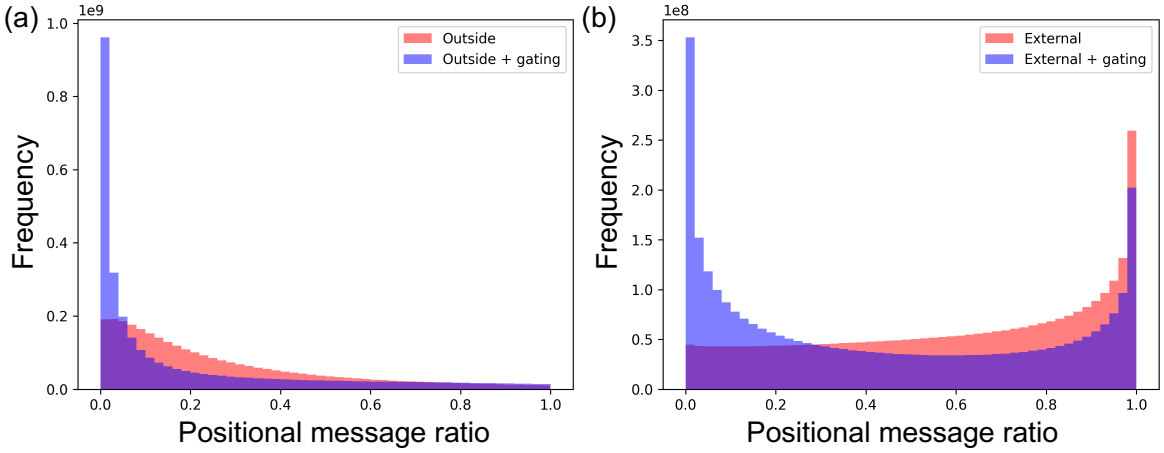

Figure 3. Histograms of the positional message ratio for the outside and external formulations, with and without gating. In the outside formulation, the ratio is concentrated at 0.0, whereas in the external formulation it is concentrated at 1.0. Gating shifts the outside distribution toward 0.0, while the external distribution becomes bimodal, with peaks near 0.0 and 1.0.

from the choice of positional encoding. Whereas prior architectures often changed these two components jointly, we fixed the positional encoding function and varied only the attention formulation, allowing the effect of attention design itself to be isolated. Within the shared geometric setting, vector positional encoding was most effective when injected inside the softmax, suggesting that geometric information is especially useful when it can directly influence attention routing rather than only perturb the transmitted message. We further found that the best-performing configuration combines an external geometric pathway with gating, as the external term allows the model to incorporate messages that are entirely different from those in the main branch.

Several directions remain open for future work. First, our study focused on formulations built on SE(3)-invariant atom-wise embeddings, and did not address settings in which the token representations themselves are SE(3)-equivariant, as in Equiformer (Liao & Smidt, 2023). Extending the comparison of formulations to these equivariant models is an important step toward a broader understanding of attention design in molecular modeling. Second, our work fixed the positional encoding function to isolate the role of attention formulation; the natural complementary direction is therefore to conduct an apples-to-apples comparison of positional encoding under a fixed attention formulation.

Acknowledgements

Y.I. is partly supported by JST SPRING, Grant Number JPMJSP2138.

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 of which we feel must be specifically highlighted here.

References

Ansel, J., Yang, E., He, H., Gimelshein, N., Jain, A., Voznesensky, M., Bao, B., Bell, P., Berard, D., Burovski, E., Chauhan, G., Chourdia, A., Constable, W., Desmaison, A., DeVito, Z., Ellison, E., Feng, W., Gong, J., Gschwind, M., Hirsh, B., Huang, S., Kalambarkar, K., Kirsch, L., Lazos, M., Lezcano, M., Liang, Y., Liang, J., Lu, Y., Luk, C. K., Maher, B., Pan, Y., Puhrsch, C., Reso, M., Saroufim, M., Siraichi, M. Y., Suk, H., Zhang, S., Suo, M., Tillet, P., Zhao, X., Wang, E., Zhou, K., Zou, R., Wang, X., Mathews, A., Wen, W., Chanan, G., Wu, P., and Chintala, S. PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, ASPLOS '24, pp. 929–947, 2024. URL http://dx.doi.org/10.1145/3620665.3640366.

Aykent, S. and Xia, T. GotenNet: Rethinking Efficient 3D Equivariant Graph Neural Networks. In *The Thirteenth International Conference on Learning Representations (ICLR 2025)*, 2025. URL https://openreview.net/forum?id=5wxCQDtbMo.

Batatia, I., Kovacs, D. P., Simm, G. N. C., Ortner, C.,

and Csanyi, G. MACE: Higher Order Equivariant Message Passing Neural Networks for Fast and Accurate Force Fields. In *Advances in Neural Information Processing Systems (NeurIPS 2022)*, 2022. URL https://openreview.net/forum?id=YPpSngE-ZU.

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. *Nat. Commun.*, 13(2453), 2022. URL http://dx.doi.org/10.1038/s41467-022-29939-5.

Catacutan, D. B., Alexander, J., Arnold, A., and Stokes, J. M. Machine learning in preclinical drug discovery. *Nat. Chem. Biol.*, 20:960–973, 2024. URL http://dx.doi.org/10.1038/s41589-024-01679-1.

Chen, C. and Ong, S. P. A universal graph deep learning interatomic potential for the periodic table. *Nat. Comput. Sci.*, 2(11):718–728, 2022. URL https://doi.org/10.1038/s43588-022-00349-3.

Chen, C., Ye, W., Zuo, Y., Zheng, C., and Ong, S. P. Graph Networks as a Universal Machine Learning Framework for Molecules and Crystals. *Chem. Mater.*, 31(9):3564–3572, 2019. URL https://doi.org/10.1021/acs.chemmater.9b01294.

Choudhary, K. and DeCost, B. Atomistic Line Graph Neural Network for improved materials property predictions. *npj Comput. Mater.*, 7(1):185, 2021. URL https://doi.org/10.1038/s41524-021-00650-1.

Choukroun, Y. and Wolf, L. Geometric Transformer for End-to-End Molecule Properties Prediction. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI 2022)*, 2022. URL https://www.ijcai.org/proceedings/2022/0401.pdf.

Dey, R. and Salem, F. M. Gate-variants of Gated Recurrent Unit (GRU) neural networks. In *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 1597–1600, 2017. URL http://dx.doi.org/10.1109/MWSCAS.2017.8053243.

Gasteiger, J., Groß, J., and Günnemann, S. Directional Message Passing for Molecular Graphs. In *International Conference on Learning Representations (ICLR 2020)*, 2020. URL https://openreview.net/forum?id=B1eWbxStPH.

Hohenberg, P. and Kohn, W. Inhomogeneous Electron Gas. *Phys. Rev.*, 136(3B):B864–B871, 1964. URL http://dx.doi.org/10.1103/PhysRev.136.B864.

Jacobs, R., Morgan, D., Attarian, S., Meng, J., Shen, C., Wu, Z., Xie, C. Y., Yang, J. H., Artrith, N., Blaiszik, B., Ceder, G., Choudhary, K., Csanyi, G., Cubuk, E. D., Deng, B., Drautz, R., Fu, X., Godwin, J., Honavar, V., Isayev, O., Johansson, A., Kozinsky, B., Martiniani, S., Ong, S. P., Poltavsky, I., Schmidt, K., Takamoto, S., Thompson, A. P., Westermayr, J., and Wood, B. M. A practical guide to machine learning interatomic potentials – Status and future. *Curr. Opin. Solid State Mater. Sci.*, 35:101214, 2025. URL http://dx.doi.org/10.1016/j.cossms.2025.101214.

Kingma, D. P. and Ba, J. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations (ICLR 2015)*, 2015. URL http://arxiv.org/abs/1412.6980.

Kohn, W. and Sham, L. J. Self-Consistent Equations Including Exchange and Correlation Effects. *Phys. Rev.*, 140(4A):A1133–A1138, 1965. URL http://dx.doi.org/10.1103/PhysRev.140.A1133.

Kwak, B., Park, J., Kang, T., Jo, J., Lee, B., and Yoon, S. GeoT: A Geometry-Aware Transformer for Reliable Molecular Property Prediction and Chemically Interpretable Representation Learning. *ACS Omega*, 8(42):39759–39769, 2023. URL http://dx.doi.org/10.1021/acsomega.3c05753.

Liao, Y.-L. and Smidt, T. Equiformer: Equivariant Graph Attention Transformer for 3D Atomistic Graphs. In *The Eleventh International Conference on Learning Representations (ICLR 2023)*, 2023. URL https://openreview.net/forum?id=KwmPfARgOTD.

Liao, Y.-L., Wood, B. M., Das, A., and Smidt, T. EquiformerV2: Improved Equivariant Transformer for Scaling to Higher-Degree Representations. In *The Twelfth International Conference on Learning Representations (ICLR 2024)*, 2024. URL https://openreview.net/forum?id=mCOBKZmrzD.

Liu, Y., Wang, L., Liu, M., Lin, Y., Zhang, X., Oztekin, B., and Ji, S. Spherical Message Passing for 3D Molecular Graphs. In *International Conference on Learning Representations (ICLR 2022)*, 2022. URL https://openreview.net/forum?id=givsRXsOt9r.

Luo, S., Chen, T., Xu, Y., Zheng, S., Liu, T.-Y., Wang, L., and He, D. One Transformer Can Understand Both 2D & 3D Molecular Data. In *The Eleventh International Conference on Learning Representations (ICLR 2023)*, 2023. URL https://openreview.net/forum?id=vZTp1oPV3PC.

Park, C. W. and Wolverton, C. Developing an improved crystal graph convolutional neural network framework

for accelerated materials discovery. *Phys. Rev. Mater.*, 4: 063801, 2020. URL https://doi.org/10.1103/PhysRevMaterials.4.063801.

Pollice, R., dos Passos Gomes, G., Aldeghi, M., Hickman, R. J., Krenn, M., Lavigne, C., Lindner-D'Addario, M., Nigam, A., Ser, C. T., Yao, Z., and Aspuru-Guzik, A. Data-Driven Strategies for Accelerated Materials Design. *Acc. Chem. Res.*, 54(4):849–860, 2021. URL https://doi.org/10.1021/acs.accounts.0c00785.

Qiu, Z., Wang, Z., Zheng, B., Huang, Z., Wen, K., Yang, S., Men, R., Yu, L., Huang, F., Huang, S., Liu, D., Zhou, J., and Lin, J. Gated Attention for Large Language Models: Non-linearity, Sparsity, and Attention-Sink-Free. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems (NeurIPS 2025)*, 2025. URL https://openreview.net/forum?id=1b7whO4SfY.

Ramakrishnan, R., Dral, P. O., Rupp, M., and von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Sci. Data*, 1(1), 2014. URL http://dx.doi.org/10.1038/sdata.2014.22.

Schütt, K., Kindermans, P.-J., Sauceda Felix, H. E., Chmiela, S., Tkatchenko, A., and Müller, K.-R. SchNet: A continuous-filter convolutional neural network for modeling quantum interactions. In *Advances in Neural Information Processing Systems (NeurIPS 2017)*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/303ed4c69846ab36c2904d3ba8573050-Paper.pdf.

Schütt, K., Unke, O., and Gastegger, M. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In *Proceedings of the 38th International Conference on Machine Learning (ICML 2021)*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9377–9388, 2021. URL https://proceedings.mlr.press/v139/schutt21a.html.

Shi, Y., Zheng, S., Ke, G., Shen, Y., You, J., He, J., Luo, S., Liu, C., He, D., and Liu, T.-Y. Benchmarking Graphormer on Large-Scale Molecular Modeling Datasets, 2023. URL https://arxiv.org/abs/2203.04810.

Spearman, C. The Proof and Measurement of Association between Two Things. *Am. J. Psychol.*, 15(1):72–101, 1904. URL http://dx.doi.org/10.2307/1412159.

Thölke, P. and Fabritiis, G. D. Equivariant Transformers for Neural Network based Molecular Potentials. In *International Conference on Learning Representations (ICLR 2022)*, 2022. URL https://openreview.net/forum?id=zNHzqZ9wrRB.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is All you Need. In *Advances in Neural Information Processing Systems (NeurIPS 2017)*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Wang, Y., Li, S., Wang, T., Shao, B., Zheng, N., and Liu, T.-Y. Geometric Transformer with Interatomic Positional Encoding. In *Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS 2023)*, 2023. URL https://openreview.net/forum?id=9o6KQrklrE.

Wood, B. M., Dzamba, M., Fu, X., Gao, M., Shuaibi, M., Barroso-Luque, L., Abdelmaqsoud, K., Gharakhanyan, V., Kitchin, J. R., Levine, D. S., Michel, K., Sriram, A., Cohen, T., Das, A., Sahoo, S. J., Rizvi, A., Ulissi, Z. W., and Zitnick, C. L. UMA: A Family of Universal Models for Atoms. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems (NeurIPS 2025)*, 2025. URL https://openreview.net/forum?id=SvopaNxYWt.

Xie, T. and Grossman, J. C. Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties. *Phys. Rev. Lett.*, 120:145301, 2018. URL https://link.aps.org/doi/10.1103/PhysRevLett.120.145301.

Xu, Z., Luo, Y., Zhang, X., Xu, X., Xie, Y., Liu, M., Dickerson, K., Deng, C., Nakata, M., and Ji, S. Molecule3D: A Benchmark for Predicting 3D Geometries from Molecular Graphs, 2021. URL https://arxiv.org/abs/2110.01717.

Ying, C., Cai, T., Luo, S., Zheng, S., Ke, G., He, D., Shen, Y., and Liu, T.-Y. Do Transformers Really Perform Badly for Graph Representation? In *Advances in Neural Information Processing Systems (NeurIPS 2021)*, volume 34, pp. 28877–28888, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/f1c1592588411002af340cbaedd6fc33-Paper.pdf.

A. Related work

Molecular property prediction has been studied extensively, and the introduction of CGCNN (Xie & Grossman, 2018) led to the rapid development of GNN-based methods for this task. In contrast to transformers, GNNs offer greater flexibility in the design of the message-passing mechanism itself. As a result, early models such as SchNet (Schütt et al., 2017) and MEGNet (Chen et al., 2019) mainly focused on pairwise interactions, while subsequent work progressively incorporated higher-order geometry through architectures capable of modeling three-body interactions, such as iCGCNN (Park & Wolverton, 2020), DimeNet (Gasteiger et al., 2020), ALIGNN (Choudhary & DeCost, 2021), and M3GNet (Chen & Ong, 2022). Moreover, SphereNet (Liu et al., 2022) further enriched geometric message passing by incorporating not only distances and angles but also torsional information through spherical message passing, thereby enabling a more expressive representation of local atomic environments.

Following the success of transformers in language modeling (Vaswani et al., 2017), a growing body of work has adapted them to molecular property prediction. Early approaches mainly relied on distance-based positional encodings. Graphormer (Ying et al., 2021) introduced a scalar positional encoding based on bond connectivity and bond-type information, which is incorporated through an inside-softmax formulation. They later proposed a Gaussian-based scalar positional encoding with the same architecture (Shi et al., 2023). Geometric Transformer (Choukroun & Wolf, 2022) instead employed a scalar positional encoding that maps interatomic distance to a scalar via an FFN and applied it outside the softmax. Subsequently, GeoT (Kwak et al., 2023) introduced a vector positional encoding by concatenating a Gaussian radial basis expansion (Eq. (10)) with the pairwise atom feature $\boldsymbol{h}_i + \boldsymbol{h}_j$, and injecting the resulting representation inside the softmax. Transformer-M (Luo et al., 2023) further extended scalar positional encoding to incorporate not only three-dimensional geometry but also two-dimensional structural information such as bond features, and injected it inside the softmax.

More recent work has moved beyond distance-only encodings by incorporating directional and angular information. Geoformer (Wang et al., 2023) was an early example in this direction: it aggregates relative direction vectors from neighboring atoms to construct atom-wise directional features. Their inner products are then used as an inside-softmax vector positional encoding, together with a cosine cutoff scalar encoding function in the outside-softmax formulation. GotenNet (Aykent & Xia, 2025) later extended this idea to higher-order directional (steerable) features and used the resulting vector positional encoding both inside the softmax and in an external pathway, gated by the target atom-wise embedding. It also employed a cosine cutoff function as the scalar encoding used in the outside-softmax and external pathway.

As summarized in Table 4, prior molecular transformer models often varied both the form of positional encoding and the way it was incorporated into attention, making it difficult to disentangle their respective contributions. Importantly, these architectures could be interpreted as specific design points within the general formulation shown in Eq. (9). From this perspective, our experiments did not amount to a collection of isolated model-to-model comparisons; rather, they provided a controlled evaluation within a common formulation space that encompassed major prior design choices.

As mentioned in Sec. 6, our study focused on models in which atom-wise embeddings are treated as SE(3)-invariant. In contrast, there exists another line of work in which the atomic representations themselves are treated as SE(3)-equivariant. Early examples include PaiNN (Schütt et al., 2021) and TorchMD-NET (Thölke & Fabritiis, 2022), which introduce equivariant vector features through interactions based on relative position vectors. Subsequent models use high-order steerable features that enable relative position vectors to interact with atom-wise features without reducing them to SE(3)-invariant quantities. These include NequIP (Batzner et al., 2022), a GNN, and Equiformer (Liao & Smidt, 2023), an equivariant transformer. EquiformerV2 (Liao et al., 2024) further significantly reduced the computational cost by designing

Table 4. Summary of attention formulations in prior molecular transformer models. Here, ϕ and $\boldsymbol{\psi}$ denote scalar and vector positional encodings, respectively.

Method	ϕ^{in}	ϕ^{out}	ϕ^{ext}	$\boldsymbol{\psi}^{\text{in}}$	$\boldsymbol{\psi}^{\text{out}}$	$\boldsymbol{\psi}^{\text{ext}}$	**Gate**	**Positional encoding description**
Graphormer (Ying et al., 2021)	✓						–	Shortest-path and bond-type scalar bias
Graphormer-3D (Shi et al., 2023)	✓						–	Gaussian scalar bias from 3D distances
Geometric Transformer (Choukroun & Wolf, 2022)		✓					–	Distance-based scalar encoding via FFN
GeoT (Kwak et al., 2023)				✓			–	GBF vector encoding with pair features
Transformer-M (Luo et al., 2023)	✓						–	Scalar bias from 2D structure and 3D geometry
Geoformer (Wang et al., 2023)		✓		✓			–	Directional features with cosine cutoff
GotenNet (Aykent & Xia, 2025)		✓	✓	✓		✓	$\boldsymbol{h}_j$	Steerable directional features with cosine cutoff

Table 5. Detailed network architecture.

Blocks	Output dims	Details
Atom embedding	$(256, N)$	Initial atom embeddings calculated by Eq. (12)
Self-attention block $\times$ 8	$(256, N)$	Model dim d: 256 Number of heads: 8 Query/Key/Value dims: 32 FFN: Linear-ReLU-Linear (dims: $256 \to 1024 \to 256$) Activation func: ReLU
Pooling	256	Mean pooling
Feed forward	1	FFN: Linear-ReLU-Linear (dims: $256 \to 256 \to 1$)

the choice of coordinate system. Exploring optimal formulations for these models would also be an interesting direction for future work.

B. Model architecture and training details

Overall architecture. Our network architecture largely follows the original transformer architecture (Vaswani et al., 2017), and Table 5 summarizes its details. After the atom-wise embeddings are initialized in $\mathbb{R}^d$ ($d = 256$), as described in the next paragraph, the representations are processed by an 8-layer transformer with 8-head multi-head attention. A molecule-level representation is then obtained by mean-pooling the final atom representations. This pooled representation is passed through a three-layer feed-forward network with dimensions $256 \to 256 \to 1$. The hidden layer uses a ReLU activation function, and the final layer predicts the scalar target property. The model was implemented using PyTorch (Ansel et al., 2024).

Embedding initialization. Following GotenNet (Aykent & Xia, 2025), we design the initialization layer such that each atom is represented by two sources of information: its atomic species and the relative positional information of its neighboring atoms. Specifically, the atomic number of atom i z_i is first mapped to a learnable embedding: $\boldsymbol{h}_i^{\text{atom}} = \text{Emb}_{\text{atom}}(z_i) \in \mathbb{R}^d$. For each pair (i, j), we compute the interatomic distance $d_{ij} = \|\boldsymbol{p}_i - \boldsymbol{p}_j\|_2$, and encode the relative positional information using a radial basis expansion with a smooth cutoff: $\boldsymbol{e}_{ij} = f_{\text{cut}}(d_{ij})\, \boldsymbol{d}_{\text{rbf}}$, where $f_{\text{cut}}(\cdot) \in \mathbb{R}$ denotes the cosine cutoff function and $\boldsymbol{d}_{\text{rbf}} \in \mathbb{R}^d$ denotes the Gaussian radial basis positional encoding described in Sec. 4.1. The geometric environment of atom i is then aggregated by:

$$\boldsymbol{g}_i = \sum_j \boldsymbol{e}_{ij} \odot \boldsymbol{h}_j^{\text{atom}}, \tag{11}$$

where $\odot$ denotes element-wise multiplication. Finally, the initialized representation is obtained by combining the central atomic embedding with the aggregated neighborhood descriptor $\boldsymbol{g}_i$ through a normalized nonlinear projection:

$$\boldsymbol{h}_i^{(0)} = W_{\text{out}}\, \sigma\big(\text{LN}\big(W_{\text{hid}}[\boldsymbol{g}_i \,\|\, \boldsymbol{h}_i^{\text{atom}}]\big)\big)\,, \tag{12}$$

where $\|$ denotes concatenation, LN denotes layer normalization, $\sigma(\cdot)$ is the SiLU activation function, and $W_{\text{hid}} \in \mathbb{R}^{d\times 2d}$ and $W_{\text{out}} \in \mathbb{R}^{d\times d}$ are trainable matrices.

Training. The model was trained using the mean absolute error loss function and the Adam optimizer (Kingma & Ba, 2015) with a batch size of 256. We employed a cosine decay learning rate schedule with a 6-epoch warmup. The maximum learning rate was set to 3×10^{-4} for isotropic polarizability (α), HOMO–LUMO gap ($\Delta\epsilon$), and dipole moment (μ), and to 1×10^{-4} for internal energy at 0 K (U_0) and zero point vibrational energy (*ZPVE*). Training was performed for 600 epochs in total.

C. Definition and interpretation of aggregate metrics

Let $\mathcal{T} = \{1, \ldots, T\}$ denote the set of evaluation tasks, and let MAE_t denote the mean absolute error on task $t \in \mathcal{T}$. To summarize performance across tasks with different target scales, we report two aggregate metrics: nMAE, reported as ***std.***, and lgMAE, reported as ***log***.

The first metric, nMAE, averages task-wise MAEs after normalizing each by the empirical standard deviation of the

corresponding target values:

$$\mathrm{nMAE} = \frac{1}{T}\sum_{t=1}^{T}\frac{\mathrm{MAE}_t}{s_t}, \tag{13}$$

where s_t is the empirical standard deviation of the target values for task t.

The second metric, lgMAE, is the average logarithm of the task-wise MAEs:

$$\mathrm{lgMAE} = \frac{1}{T}\sum_{t=1}^{T}\log\left(\mathrm{MAE}_t\right) = \log\left[\left(\prod_{t=1}^{T}\mathrm{MAE}_t\right)^{1/T}\right]. \tag{14}$$

Please note that lgMAE can be viewed as the logarithm of the geometric mean of the task-wise MAEs.

These two metrics provide complementary summaries of cross-task performance. The normalized metric nMAE accounts for heterogeneity in target scales by expressing each task's MAE in units of the empirical standard deviation of its target values. It can therefore be interpreted as the average prediction error relative to the empirical variability of each task.

In contrast, lgMAE aggregates task-wise MAEs through their geometric mean, up to a logarithmic transformation. As a result, it reflects the typical multiplicative scale of the errors rather than their arithmetic average. An arithmetic mean can be dominated by a small number of tasks with very large MAEs, since errors are combined additively on the original scale. The geometric mean instead combines errors multiplicatively, or equivalently additively in log space, which compresses the effect of unusually large errors. Thus, lgMAE provides a summary of typical task-wise absolute error that is less sensitive to high-error outlier tasks.

D. Results with a different positional encoding function

In Secs. 4.2 and 4.3, we identified two main findings: (i) for vector positional encoding, the inside-softmax formulation outperforms the outside-softmax formulation; and (ii) the external term is most effective when combined with value-vector gating.

To test whether these observations generalize beyond a single positional encoding, we replaced the vector positional encoding function and performed controlled experiments. We consider a new vector positional encoding inspired by GeoT (Kwak et al., 2023). Using the Gaussian radial basis function $\boldsymbol{b}_{ij}$ from Eq. (10) and atom-wise embeddings $\boldsymbol{h}_i^{(\ell)}$, we define

$$\boldsymbol{\psi}_{ij} = W\left(\left[W_{\mathrm{r}}\boldsymbol{b}_{ij} \parallel \left(\boldsymbol{h}_i^{(\ell)} + \boldsymbol{h}_j^{(\ell)}\right)\right]\right), \tag{15}$$

where $\parallel$ denotes concatenation, $W \in \mathbb{R}^{d\times 2d}$, and $W_{\mathrm{r}} \in \mathbb{R}^{d\times K}$ are learnable parameters. This positional encoding function adds parameter matrices of sizes 256×512 and 256×64 for each vector positional encoding across 8 layers, corresponding to 1.18M additional parameters in total.

To examine the first finding, we fixed the scalar positional encoding to the *inside* formulation and compared alternative placements of the vector positional encoding as shown in Table 6 (I). To examine the role of the external term and value gating, we compared four additional variants in Table 6 (II): *inside* + *outside* and *inside* + *external*, each with and without value gating. For the outside and external terms, we used only the original Gaussian radial basis function when combined with the gating mechanism, since otherwise the gating mechanism would interact with information overlapping with the newly introduced atom-wise embeddings.

The results are consistent with those in Secs. 4.2 and 4.3. The inside formulation for vector positional encoding outperforms the outside-softmax formulation overall. In addition, while the external formulation alone is not competitive, combining it with value gating yields the best performance. Taken together, these results indicate that our conclusions are not tied to a particular positional encoding design.

E. Results with a different initialization of atom-wise embeddings

In Appendix D, we confirmed that the main findings are also observed when using a different positional encoding. We further investigated whether the results can be reproduced when the initialization of atom-wise embeddings is changed. In

Table 6. Results of a controlled study using an alternative GeoT-inspired vector positional encoding. The first block compares different placements of the vector positional encoding with the scalar positional encoding fixed to the inside formulation. The second block evaluates inside + outside and inside + external variants with and without value gating. In each subsection, the best and second-best results are highlighted in bold and with underlining, respectively.

#	ϕ^{in}	ϕ^{out}	ϕ^{ext}	ψ^{in}	ψ^{out}	ψ^{ext}	Gate	Added param (Million)	α (ma_0^3)	$\Delta\epsilon$ (meV)	μ (mD)	U_0 (meV)	ZPVE (meV)	std. (%)	log -
0							–	0 (+0.0%)	71.31	54.60	116.9	12.62	1.617	2.635	-3.699
(I) Comparison of vector positional encodings															
1	✓						–	0 (+0.0%)	67.53	48.26	47.04	**12.53**	1.563	1.599	-3.925
2	✓			✓			–	1.18 (+12.1%)	**61.01**	43.56	**36.70**	13.12	1.551	**1.373**	**-4.008**
3	✓				✓		–	1.18 (+12.1%)	66.54	47.07	45.78	12.70	1.564	1.562	-3.935
4	✓			✓	✓		–	2.36 (+24.2%)	68.12	**42.53**	40.45	12.60	**1.524**	1.423	-3.982
(II) Comparison of external positional encodings and gating															
4	✓			✓	✓		–	2.36 (+24.2%)	68.12	42.53	40.45	12.60	1.524	1.423	-3.982
11	✓			✓	✓		$\boldsymbol{v}_j$	2.88 (+29.6%)	59.16	43.29	35.85	12.17	1.495	1.350	-4.042
14	✓		✓	✓		✓	–	2.36 (+24.2%)	93.63	49.03	55.49	16.47	1.823	1.800	-3.738
15	✓		✓	✓		✓	$\boldsymbol{v}_j$	2.88 (+29.6%)	**53.73**	**39.01**	**32.04**	**10.63**	**1.351**	**1.214**	**-4.152**

Table 7. Results of a controlled study using an alternative initialization of atom-wise embeddings. The first block compares different placements of the vector positional encoding with the scalar positional encoding fixed to the inside formulation. The second block evaluates inside + outside and inside + external variants with and without value gating.

#	ϕ^{in}	ϕ^{out}	ϕ^{ext}	ψ^{in}	ψ^{out}	ψ^{ext}	Gate	Added param (Million)	α (ma_0^3)	$\Delta\epsilon$ (meV)	μ (mD)	U_0 (meV)	ZPVE (meV)	std. (%)	log -
0							–	0 (+0.0%)	6340	1076	1155	917.2	40.15	50.23	-0.247
(I) Comparison of vector positional encodings															
1	✓						–	0 (+0.0%)	185.0	70.96	104.4	93.48	6.521	3.267	-2.799
2	✓			✓			–	0.13 (+1.4%)	56.94	**40.65**	**40.26**	12.38	1.438	1.362	-4.043
3	✓				✓		–	0.13 (+1.4%)	61.58	45.03	50.80	13.83	1.739	1.590	-3.901
4	✓			✓	✓		–	0.26 (+2.8%)	**54.73**	41.05	40.31	**10.83**	**1.420**	**1.360**	**-4.078**
(II) Comparison of external positional encodings and gating															
4	✓			✓	✓		–	0.26 (+2.8%)	54.73	41.05	40.31	10.83	1.420	1.360	-4.078
11	✓			✓	✓		$\boldsymbol{v}_j$	0.79 (+8.3%)	55.66	38.96	36.53	10.26	1.448	1.279	-4.112
14	✓		✓	✓		✓	–	0.26 (+2.8%)	56.09	39.81	35.48	12.19	1.388	1.282	-4.086
15	✓		✓	✓		✓	$\boldsymbol{v}_j$	0.79 (+8.3%)	**52.66**	**36.98**	**32.24**	**9.526**	**1.341**	**1.180**	**-4.189**

the previous experiments, Eq. (12) was used, where positional information was already included at the initialization stage. Therefore, as an alternative initialization method, we conducted the same experiments using embeddings initialized solely based on atomic species. Specifically, each atomic species is assigned a trainable embedding, which is used as the initial embedding. This is equivalent to applying one-hot encoding followed by a learnable projection into our feature dimension d.

Table 7 (I) presents the comparison results for vector positional encoding, while Table 7 (II) shows the results for external positional encoding and gating. Consistent with the findings in Secs. 4.2 and 4.3, we observe that the inside variant performs well for vector positional encoding, and that the combination of external + gating encoding performs the best. These results demonstrate that the same conclusions hold regardless of the embedding initialization.

F. Results on a different dataset

In this study, we used the QM9 dataset (Ramakrishnan et al., 2014). Since QM9 contains relatively small molecules with atomic numbers up to 9, we additionally conducted experiments on the much larger Molecule3D dataset (Xu et al., 2021) to examine whether our main findings also hold for a larger and more diverse dataset. Molecule3D contains 3,899,647 molecular structures, making it more than 29 times larger than the QM9 dataset. In addition, the average number of atoms per molecule is approximately 1.6 times greater, and the dataset includes heavier elements. As the target property, we used the HOMO–LUMO gap. Following prior work (Wang et al., 2023), we evaluated our method under two data splits: Random

Table 8. Results of a controlled study on the Molecule3D dataset (HOMO–LUMO gap). The first block compares different placements of the vector positional encoding with the scalar positional encoding fixed to the inside formulation. The second block evaluates inside + outside and inside + external variants with and without value gating. In each subsection, the best and second-best results are highlighted in bold and with underlining, respectively.

#	ϕ^{in}	ϕ^{out}	ϕ^{ext}	$\boldsymbol{\psi}^{\text{in}}$	$\boldsymbol{\psi}^{\text{out}}$	$\boldsymbol{\psi}^{\text{ext}}$	Gate	Added param (Million)	Random (meV)	Scaffold (meV)
0							–	0 (+0.0%)	40.89	212.8
(I) Comparison of vector positional encodings										
1	✓						–	0 (+0.0%)	32.42	181.2
2	✓			✓			–	0.03 (+0.59%)	<u>25.50</u>	<u>141.1</u>
3	✓				✓		–	0.03 (+0.59%)	29.14	159.1
4	✓			✓	✓		–	0.06 (+1.18%)	**24.84**	**133.0**
(II) Comparison of external positional encodings and gating										
4	✓			✓	✓		–	0.06 (+1.18%)	24.84	133.0
11	✓			✓	✓		$\boldsymbol{v}_j$	0.33 (+5.90%)	<u>24.06</u>	<u>129.6</u>
14	✓		✓	✓		✓	–	0.06 (+1.18%)	25.60	134.0
15	✓		✓	✓		✓	$\boldsymbol{v}_j$	0.33 (+5.90%)	**24.00**	**127.5**

and Scaffold. The Random split is obtained by randomly partitioning the data, whereas the Scaffold split is more challenging because it introduces structural bias between the training and test sets. For both splits, the numbers of training, validation, and test samples were 2,339,788, 779,929, and 779,930, respectively.

As in Secs. 4.2 and 4.3, we used a cosine cutoff function for the scalar positional encoding and Gaussian radial basis functions for the vector positional encoding. To enable training on this large-scale dataset, we reduced the number of transformer layers to 4 and the dimension of the Gaussian radial basis expansion (K in Eq. (10)) to 32. We used a batch size of 256 and optimized the mean absolute error with the Adam optimizer, using a learning rate of 3×10^{-4}, a cosine decay scheduler, and 6 epochs of warmup.

Table 8 (I) presents the comparison results for vector positional encoding, and Table 8 (II) shows the results for external positional encoding and gating. These results indicate that the main findings in Secs. 4.2 and 4.3 hold not only on the relatively small QM9 dataset, but also on the large-scale Molecule3D dataset.

G. Effect of the external positional encoding term on which atoms interact strongly

To examine whether the atoms attended to differ between messages derived from value vectors and those derived from positional encodings, we computed the norm of each message and evaluated the Spearman's rank correlation coefficient (Spearman, 1904) between them. Specifically, the value vector-derived message is defined as $\boldsymbol{l}^{\text{v}}_{ij} = \alpha^{(\ell)}_{ij}\boldsymbol{v}^{(\ell)}_j$, whereas the positional encoding-derived message is defined as $\boldsymbol{l}^{\text{p}}_{ij} = \alpha^{(\ell)}_{ij}\boldsymbol{\psi}^{\text{out}}_{ij}$ in the outside formulation, and $\boldsymbol{l}^{\text{p}}_{ij} = \phi^{\text{ext}}_{ij}\boldsymbol{\psi}^{\text{ext}}_{ij}$ in the external formulation. For each atom i, we computed Spearman's correlation coefficient by sorting $||\boldsymbol{l}^{\text{v}}_{ij}||_2$ and $||\boldsymbol{l}^{\text{p}}_{ij}||_2$ over j, then computing the Pearson correlation between their ranks using the top five entries. The coefficient is 1 when the rankings are identical and -1 when they are exactly reversed.

Figure 4 shows histograms for the *outside + gating* and *external + gating* settings, constructed from all layers and heads using 1,000 randomly selected test samples from each task. In the outside formulation, the rankings of message magnitudes derived from value vectors and positional encodings are almost aligned. In contrast, under the external formulation, the two rankings are often weakly correlated or uncorrelated. This suggests that, as discussed in Sec. 5.2, the external formulation enables interactions with atoms that differ from those in the main branch.

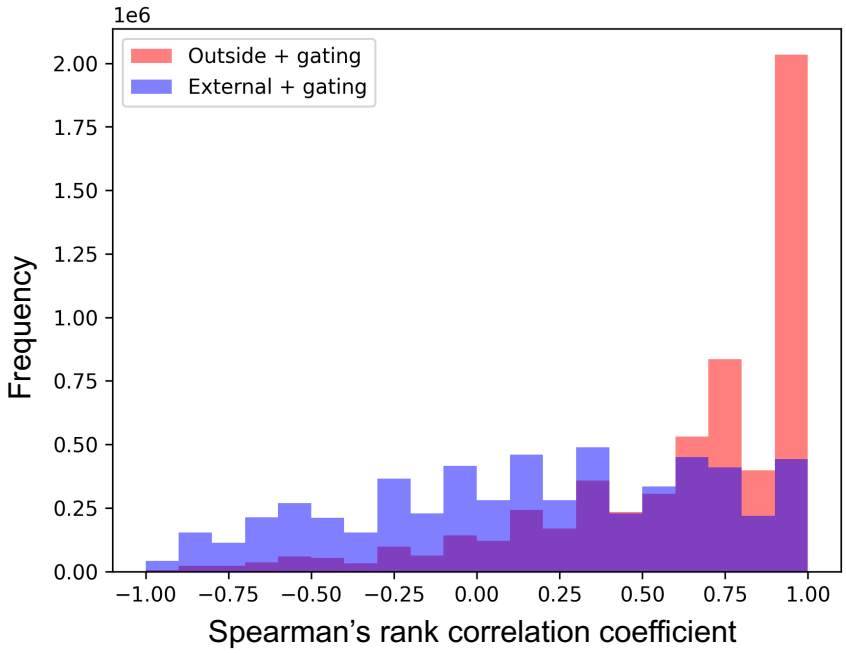

Figure 4. Histograms of the Spearman's rank correlation coefficients between the rankings of message norms from value vectors and positional encodings, computed over all layers and heads. The outside + gating setting shows strong agreement between the two rankings, whereas the external + gating setting often exhibits weak or no correlation, suggesting that the external formulation interacts with atoms different from those in the main branch.