SPATIAL ATTENTION KINETIC NETWORK WITH E(N)-EQUIVARIANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural networks that are equivariant to rotations, translations, reflections, and permutations on n-dimensional geometric space have shown promise in physical modeling for tasks such as accurately but inexpensively modeling complex potential energy surfaces to guiding the sampling of complex dynamical systems or forecasting their time evolution. Current state-of-the-art methods employ spherical harmonics to encode higher-order interactions among particles, which are computationally expensive. In this paper, we propose a simple alternative functional form that uses neurally parametrized linear combinations of edge vectors to achieve equivariance while still universally approximating node environments. Incorporating this insight, we design *spatial attention kinetic networks* with E(n)-equivariance, or SAKE, which are competitive in many-body system modeling tasks while being significantly faster.

1 INTRODUCTION

Encoding the relevant symmetries of systems of interest into the inductive biases of deep learning architectures has been shown to be crucial in physical and chemical modeling. Graph neural networks (GNNs) (Kipf and Welling, 2016; Xu et al., 2018; Gilmer et al., 2017; Hamilton et al., 2017; Battaglia et al., 2018), for instance, preserve permutation equivariance by applying indexing-invariant pooling functions among nodes (particles) and edges (pair-wise interactions) and have emerged to become a powerful workhorse in a wide range of modeling tasks for many-body system (Satorras et al., 2021).

When it comes to describing not only the *topology* of the system but also the *geometry* of the state, relevant symmetry groups for three-dimensional systems are SO(3) (rotational equivariance), SE(3) (rotational and translational equivariance), and E(3) (additionally with reflectional equivariance). A ubiquitous and naturally invariant first attempt to encode the geometry of such systems is to employ only radial information, i.e., interparticle distances. This alone has empirically displayed utility in predicting quantum chemical potential energies, and considerable effort has been made in the fine-tuning of radial filters to achieve quantum chemical accuracy—1 kcal/mol, the empirical threshold to qualitatively reliably predict the behavior of a quantum mechanical system—and beyond (Schütt et al., 2017).

Nonetheless, radial information alone is not sufficient to fully describe node environments—the spatial distribution of neighbors around individual particles. The relative locations of particles around a central node could drastically change despite maintaining distances to these neighbors unaltered. To completely describe node environments, one needs to address these remaining degrees of freedom. Current state-of-the-art approaches encode angular distributions by employing a truncated series of spherical harmonics to generate higher-order feature representations; while these models have been shown to be data efficient for learning properties of physical systems, these features are expensive to compute, with the expense growing rapidly with the order of harmonics included (Thomas et al., 2018; Klicpera et al., 2021a; Fuchs et al., 2020; Batzner et al., 2021; Anderson et al., 2019). The prohibitive cost would prevent this otherwise performant class of models from being employed in materials and drug design, where rapid simulations of large systems are crucial to provide quantitative insights.

Here, we design a simple functional form, which we call *spatial attention*, that uses the norm of a set of neurally parametrized linear combinations of edge vectors to describe the node environment.

Though simple in form, easy to engineer, and ultra-fast to compute, spatial attention is capable of universally approximating any functions defined on local node environment while preserving E(n)-invaraince/equivariance in arbitrary n-dimensional space.

After demonstrating the approximation universality and invariance of spatial attention, we incorporate it into a novel neural network architecture that uses spatial attention repeatedly and keep track of a fictitious velocity variable, which we call a *spatial attention kinetic network with E(n)-Equivariance*, or SAKE (pronounced *sah-keh*, like the Japanese rice wine). To demonstrate the robustness and versatility of SAKE, we benchmark its performance on potential energy approximation and dynamical system forecasting and sampling tasks. For all popular benchmarks, compared to state-of-the-art models, SAKE achieves competitive performance on a wide range of invariant (MD17: Table 1, QM9: Table 3, ISO17: Table 2) and equivariant (N-body charged particle Table 4, walking motion: Table 5) while requiring only a fraction of their training and inference time. Furthermore, we incorporate SAKE into an equivariant normalizing flow framework and show its utility in exact likelihood sampling tasks.

2 BACKGROUND

In this section, we provide some theoretical background on physical modeling, equivariance, and graph neural networks to lay the groundwork for the exposition of spatial attention networks.

2.1 EQUIVARIANCE: PERMUTATIONAL, ROTATIONAL, TRANSLATIONAL, AND REFLECTIONAL

A function $f : \mathcal{X} \to \mathcal{Y}$ is said to be equivariant to a symmetry group G if

$$f(T_g(\mathbf{x})) = S_g(f(\mathbf{x})) \tag{1}$$

$\forall\, g \in G$ and some equivalent transformations on the two spaces respectively $T_g : \mathcal{X} \to \mathcal{X}$ and $S_g : \mathcal{Y} \to \mathcal{Y}$.

If on a n-dimensional space $\mathcal{X} = \mathcal{Y} = \mathbb{R}^n$, we have permutation P and $T_g(\mathbf{x}) = P\mathbf{x}, S_g(\mathbf{y}) = P\mathbf{y}$ satisfying Equation 1, we say f is permutationally equivariant; if $T_g(\mathbf{x}) = \mathbf{x}R$ where $R \in \mathbb{R}^{n\times n}$ is a rotation matrix $RR^T = I$, and $S_g(\mathbf{y}) = \mathbf{y}R$ we say f is rotationally equivariant; if $T_g(\mathbf{x}) = \mathbf{x} + \Delta\mathbf{x}$ and $S_g(\mathbf{y}) = \mathbf{y} + \Delta\mathbf{x}$, where $\mathbf{x} \in \mathbb{R}^n$ we say f is translationally equivariant; finally, if $T_g(\mathbf{x}) = \mathrm{Ref}_\theta(\mathbf{x})$ and $S_g(\mathbf{y}) = \mathrm{Ref}_\theta(\mathbf{y})$, and Ref_θ is a reflection on n-dimensional space, we say f is reflectionally equivariant.

2.2 GRAPH NEURAL NETWORKS

Modern GNNs, which exchanges and summarizes information among nodes and edges, are better analyzed through the *spatial* rather than *spectral* lens, according to Wu et al. (2019)'s classification. Following the framework from Gilmer et al. (2017); Xu et al. (2018); Battaglia et al. (2018), for a node v with neighbors $u \in \mathcal{N}(v)$, in a graph $\mathcal{G}$, with $h_v^{(k)}$ denoting the feature of node v at the k-th layer (or k-th round of message-passing) and $h_v^0 \in \mathbb{R}^D$ the initial node feature on the embedding space, the k-th message-passing step of a GNN can be written as three steps:

First, an *edge update*,

$$h_{e_{uv}}^{(k+1)} = \phi^e\big(h_u^{(k)}, h_v^{(k)}, h_{e_{uv}}^{(k)}\big), \tag{2}$$

where the feature embeddings h_u of two connected nodes u and v update their edge feature embedding $h_{e_{uv}}$, followed by *neighborhood aggregation*,

$$a_v^{(k+1)} = \rho^{e\to v}(\{h_{e_{uv}}^{(k)}, u \in \mathcal{N}(v)\}), \tag{3}$$

where edges incident to a node v pool their embeddings to form *aggregated neighbor embedding* a_v, and finally a *node update*,

$$h_v^{(k+1)} = \phi^v(a_v^{(k+1)}, h_v^{(k)}) \tag{4}$$

where $\mathcal{N}(\cdot)$ denotes the operation to return the multiset of neighbors of a node and ϕ^e and ϕ^v are implemented as feedforward neural networks. Since the neighborhood aggregation functions $\rho^{e\to v}$ are always chosen to be indexing-invariant functions, namely a SUM or a MEAN operator, Equation 3, and thereby the entire scheme, is permutationally invariant.

Problem Statement. We are interested in designing a class of parametrized functions $f_\theta : \mathcal{X} \times \mathcal{H} \to \mathcal{X} \times \mathcal{H}$ that map from and to the joint spaces of (n-dimensional) geometry $\mathcal{X} \in \mathbb{R}^n$ and semantic embedding $\mathcal{H} \in \mathbb{R}^D$ such that it is permutationally, rotationally, translationally, and reflectionally equivariant on $\mathcal{X}$ and invariant on $\mathcal{H}$. That is, for a given coördinate $\mathbf{x} \in \mathcal{X}$, embedding $h \in \mathcal{H}$ and any transformation mentioned in Section 2.1 $T : \mathcal{X} \to \mathcal{X}$ (rotation, translation, and reflection), we have:

$$\mathbf{x}_f, h_f = f_\theta(\mathbf{x}, h) \Longleftrightarrow T(\mathbf{x}_f), h_f = f_\theta(T(\mathbf{x}), h) \tag{5}$$

3 Related Work: Invariant and Equivariant GNNs

Invariant GNN. Although the notion of graph might not appear in their original publications, a plethora of neural architectures could be viewed as *de facto* graph neural networks with invariance on geometric space (for a survey, see (Han et al., 2022)). SchNet (Schütt et al., 2017), for example, approximates the potential energy of molecules by applying radially-symmetric convolutions that only operate on the distances between atoms to ensure the model is E(3)-invariant, effectively using a graph neural network architecture where nodes denote atoms and edges denote the distance separation between pairs of atoms within a cutoff radius. In addition to interatomic distances, invariant models could also incorporate information about the angular distribution of atom neighbors by computing atomic features that incorporate angular information of neighbor-atom-neighbor triplets (Smith et al., 2017; Behler, 2011; Klicpera et al., 2020a; Liu et al., 2022). Compared to SAKE, these models are not capable in *equivariant* modeling; even in an invariant setting, they are empirically less performant (See invariant tasks performance Section 6.1).

Equivariant GNN with spherical harmonics. Architectures achieving SE(3)-*equivariance* by leveraging Bessel functions and spherical harmonics to encode higher-order interactions (Thomas et al., 2018; Klicpera et al., 2021a; Fuchs et al., 2020; Anderson et al., 2019; Klicpera et al., 2021b; Brandstetter et al., 2021; Liu et al., 2022) shows outstanding performance and data efficiency, some of which is on par with SAKE (Table 1, 4). They nonetheless tend to be difficult to engineer and expensive to train and run. (See runtime benchmarks in aforementioned tables.)

Equivariant GNN with dot product scalarization. Recent efforts (Schütt et al., 2021; Thölke and Fabritiis, 2022; Huang et al., 2022) relying on dot product of edge tensors as equivariant nonlinearity have achieved competitive results on machine learning potential construction, among which none has been validated to perform well on both *equivariant* and *invariant tasks*. Only Huang et al. (2020) offered theoretical analysis regarding universality, which, we show in Remark 4, has more stringent conditions. Experimentally, they are consistently outperformed by SAKE in invariant (Section 6.1: Table 1) and equivariant (Section 6.2: Table 5) tasks.

Message passing between emebedding and geometric spaces. Satorras et al. (2021) has formalized the link between graph neural networks and and equivariance and provided a generalizing framework encompassing iterative geometric-to-embedding and embedding-to-geometric type update. Like our proposed architecture, E(N) Equivariant Graph Neural Networks (EGNN), proposed in Satorras et al. (2021), also uniquely describes the geometry of a n-body system, although their argument was based on a *global* scale given sufficient steps of message passing, while we can sufficiently describe the *local* geometric environment. This advantage is evidenced by extensive experiments (invariant: Table 3; equivariant: Table 4 and 5). In ablation study (Appendix Section 11), we show that without *spatial attention*, EGNN is not competitive in potential energy modeling, even when all other tricks used in this paper were added. Similar to EGNN (Satorras et al., 2021), our model is equivariant w.r.t. a general E(n) group and are not restricted to E(3). Also worth noting is that the Geometric Vector Perceptrons algorithm proposed in Jing et al. (2021) could be regarded as a special case of our framework where the attention weights are learned globally, whereas we learn them in an amortized manner and thus can be transductively generalized across systems.

4 Theory: Spatial Attention

Given a node v with embedding $h_v \in \mathcal{H} = \mathbb{R}^C$ (where C denotes the embedding dimension) and position $\mathbf{x}_v \in \mathcal{X} = \mathbb{R}^n$ (where n denotes the geometry dimension) in a graph $\mathcal{G}$, its neighbors

$u \in \mathcal{N}(v)$ with connecting edges $\{e_{uv}\}$, with displacement vector $\vec{\mathbf{e}}_{uv} = \mathbf{x}_v - \mathbf{x}_u$ and embedding $h_{e_{uv}} = \rho^{v \to e}(h_v, h_u)$ with some aggregation function $\rho^{v \to e}$, we define spatial attention $\phi : \mathcal{X} \times \mathcal{H} \to \mathcal{H}$ as

$$\phi^{\mathrm{SA}}(v) = \mu\big(\bigoplus_{i=1}^{N_\lambda} || \sum_{u \in \mathcal{N}(v)} \lambda_i(h_{e_{uv}}) f(\vec{\mathbf{e}}_{uv}) ||\big), \tag{6}$$

where $\lambda_i : \mathcal{H} \to \mathbb{R}^1, i = 1, ..., N_\lambda$ is a set of arbitrary attention weights-generating function that operates on the edge embedding, $f : \mathcal{X} \to \mathcal{X}$ is an *equivariant* function that operate on the edge vector, $\mu : N_\lambda \to \mathcal{H}$ is an arbitrary function that takes the norms of N_λ linear combinations, and $\oplus$ denotes concatenation. We drop the explicit dependence of $\phi^{\mathrm{SA}}(v)$ on both geometric and embedding properties of v and u for simplicity.

When implemented, λ and μ take the form of feed-forward neural networks. In other words, for each node v, we first parametrize a $N_\lambda = |\{\lambda_i\}|$ (is analogous to the number *heads* in multi-head attention) sets of attention weights based on the edge embeddings $\lambda_i(h_{e_{uv}})$. At the same time, we also equivariantly (on E(n) group) transform the edge vector into $f(\vec{\mathbf{e}}_{uv})$. Next, we use those N_λ sets of attention weights to linearly combine these vectors among edges and take the Euclidean norm of each of these linear combinations, resulting in N_λ norms. Lastly, we concatenate these norms and put them into a feed-forward neural network μ to compute the node embedding. It naturally follows that:

Remark 1. *Spatial attention is permutationally, rotationally, transalationally, and reflectionally invariant on E(n),*

since $h_{e_{uv}}$ is invariant w.r.t. indexing and the Euclidean norm function is invariant to rotation, translation, and reflection.

With all local degrees of freedom are incorporated in spatial attention, it is perhaps intuitive to see that $\phi^{\mathrm{SA}}(v)$ can uniquely define the local environment of nodes up to E(n) symmetry. We formalize this finding (Proof in Appendix Section 8.1):

Theorem 1. *For a node v in a graph $\mathcal{G}$ with neighbors $u \in \mathcal{N}(v)$ connected to u by edges with positions $\mathbf{x}_v$ and $\mathbf{x_u}$ distinct embeddings $h_{e_{vu_i}} \neq h_{e_{vu_j}}, \forall 1 \leq i, j \leq |\mathcal{N}(v)|$, and any invariant continuous function $g(h_{e_{vu_i}}, \mathbf{x}_v, \mathbf{x}_{u_i})$ on E(n) spatial attention ϕ^{SA} can approximate g with arbitrarily small error ϵ with some $\{\lambda\}$, f, and μ.*

The distinctness condition corresponds to the full-rank requirement in GMN (Huang et al., 2022). Practically, satisfying it is not difficult: we implement h_e as dependent upon edge length so even if the semantic embedding of edges are identical, the inequality $h_{e_{vu_i}} \neq h_{e_{vu_j}}$ holds as long as the system is not strictly symemtrical at all times (namely the mirror symmetry presented in hydrogen molecule). Compared to the dot product scalarization (which we generalize using the previous notation as: $Q = \bigoplus_{u \in \mathcal{N}(v)} \bigoplus_{i=1}^{N_\lambda} \lambda_{q,i}(h_{e_{uv}}) f(\vec{\mathbf{e}}_{uv}), K = \bigoplus_{u \in \mathcal{N}(v)} \bigoplus_{i=1}^{N_\lambda} \lambda_{k,i}(h_{e_{uv}}) f(\vec{\mathbf{e}}_{uv}), \phi^{\mathrm{DPS}}(v) = \mu(QK^T)$) used in GMN (Huang et al., 2022) or PaiNN (Schütt et al., 2021), though, spatial attention with the norm-taking operation is more expressive (discriminative) (Proof in Appendix Section 8.2):

Remark 2. *For two nodes v_0, v_1 in graph $\mathcal{G}$, $\phi^{\mathrm{DPS}}(v_0) \neq \phi^{\mathrm{DPS}}(v_1)$ is the necessary, but not sufficient condition that $\phi^{\mathrm{SA}}(v_0) \neq \phi^{\mathrm{SA}}(v_1)$ with some parametrization.*

5 ARCHITECTURE: SPATIAL ATTENTION KINETIC NETWORK (SAKE)

Leveraging the simple functional form introduced in Section 4, we design a fast, efficient, and easy-to-implement architecture termed a *spatial attention kinetic network with E(n)-equivariance*, or SAKE. Here, we introduce the remaining components of SAKE before assembling them in a modular way. The necessity of these designs are experimentally justified with ablation study in Appendix Section 11.

Edge embedding. To embed pairwise interactions, combine the SchNet (Schütt et al., 2017)-style continuous filter convolution and the simple concatenation of the scalar-valued distance as in Satorras et al. (2021) to achieve the balance between high radial resolution and large receptive field. The

Algorithm 1 Spatial Attention Kinetic Networks Layer

function SAKELAYER($\{h_v^{(k)}\}, \{\mathbf{x}_v^{(k)}\}, \{\mathbf{v}_v^{(k)}\}, \mathcal{G}$) ▷ Input embedding, position, and velocity
  for $v \in \mathcal{V}$ **do**
    for $u \in \mathcal{N}(v)$ **do**
      $h_{e_{uv}}^{(k)} \leftarrow \phi^e(h_v^k, h_u^k, ||\vec{\mathbf{e}}_{uv}||)$ ▷ Edge update, Sec 5 Eq 7
    end for
    $h_{e_{uv}}^{(k+1)} \leftarrow h_{e_{uv}}^{(k)} * \alpha_{uv}^{\mathcal{X}\times\mathcal{H}}$ ▷ Euclidean and semantic attention, Sec 5 Eq 8
    for $u \in \mathcal{N}(v)$ **do**
      $h_{\mathrm{SA}_v}^{(k+1)} = \phi^{\mathrm{SA}}(h_{e_{uv}}^{(k)}, \vec{\mathbf{e}}_{uv})$ ▷ Spatial attention, Sec 5 Eq 6
    end for
    $a_v^{(k)} \leftarrow \sum_{u\in\mathcal{N}(v)} h_{e_{uv}}^{(k)}$ ▷ Neighborhood aggregation, Sec 2.2 Eq 3
    $\mathbf{v}_v^{(k+1)} \leftarrow \phi^{v\to\mathcal{V}}(h_v^{(k)})\mathbf{v}^{(k)} + \mathbf{W}_v \sum_i \sum_{u\in\mathcal{N}(v)} \lambda_i(h_{e_{uv}}^{(k)}) f(\vec{\mathbf{e}}_{uv}^k)$ ▷ Vel. update, Sec 5 Eq 10
    $\mathbf{x}_v^{(K)} \leftarrow \mathbf{x}_v^{(k)} + \mathbf{v}_v^{(K)}$ ▷ Position update, Sec 5 Eq 10
    $h_v^{(k+1)} \leftarrow \phi^v(h_v^{(k)}, a_v^{(k)}, h_{\mathrm{SA}_v}^{(k)})$ ▷ Node update, Sec 2.2 Eq 4
    return $\{h_v^{(k+1)}\}, \{\mathbf{x}_v^{(k+1)}\}, \{\mathbf{v}_v^{(k+1)}\}$
  end for
end function

resulting edge embedding is thus

$$h_{e_{uv}}^{(k)} = \phi^e(h_u^{(k)} \oplus h_v^{(k)} \oplus ||\vec{\mathbf{e}}_{uv}^{(k)}|| \oplus \mathrm{RBF}(||\vec{\mathbf{e}}_{uv}^{(k)}||) \odot f^r(h_u^{(k)} \oplus h_v^{(k)})), \tag{7}$$

where $\odot$ denotes Hadamard product and f^r is a (filter-generating) feed-forward network as in Schütt et al. (2017).

Euclidean and semantic attention To promote anisotropy (which Dwivedi et al. (2020) finds useful) in the pooling operation, apart from spatial attention introduced in Section 4, we used two more types of attention mechanisms, to weigh interactions (edges) based on Euclidean and semantic distances. For a node v and a set of neighbors $\{u \in \mathcal{N}(v)\}$, positioned at $\mathbf{x}_v$ and $\mathbf{x}_u$, connected by directed edges $\vec{\mathbf{e}}_{uv}$ respectively, the Euclidean attention weights $\alpha_{u_i v}^{\mathcal{X}}$ are calculated in an additive manner (Bahdanau et al., 2014) with an adaptive scaling factor γ before $\mathrm{Softmin}$. And semantic attention weights $\alpha_{u_i v}^{\mathcal{H}}$ are calculated similar to Graph Attention Networks (Veličković et al., 2018). To produce models for the purpose of molecular simulation, To ensure continuous forces and gradients, one would need to choose as sigma σ as at least C2-continuous activation functions, namely CeLU (Barron, 2017). These weights are combined via yet another $\mathrm{Softmax}$ operation (Matveev et al., 2020):

$$\alpha_{u_i v}^{\mathcal{X}} = \frac{\exp(-\gamma||\vec{\mathbf{e}}_{u_i v}||)}{\sum_{u\in\mathcal{N}(v)} \exp(-\gamma||\vec{\mathbf{e}}_{uv}||)}; \alpha_{u_i v}^{\mathcal{H}} = \frac{\exp(\sigma(\mathbf{a}^T h_{e_{u_i v}}))}{\sum \exp(\sigma(\mathbf{a}^T h_{e_{uv}}))}; \alpha_{u_i v}^{\mathcal{X}\times\mathcal{H}} = \frac{\exp \alpha_{u_i v}^{\mathcal{X}} \alpha_{u_i v}^{\mathcal{H}}}{\sum \exp \alpha_{uv}^{\mathcal{X}} \alpha_{uv}^{\mathcal{H}}}. \tag{8}$$

It is trivial to expand to multi-head attention with k sets of $(\mathbf{a}_{1,\ldots,k}^T, \gamma_{1,\ldots,k})$ and the resulting combined attention weights concatenated.

Since the edge embedding after mixed Euclidean and semantic attention $h_{e_{uv}} * \alpha_{uv}^{\mathcal{X}\times\mathcal{H}}$ already encodes the desired inductive bias that nodes farther away from each other would have less impact on each other's embedding, we directly use this representation and simply set λ_i as a linear projection (with weights $\mathbf{W}_\lambda$) and f as identity in Equation 6:

$$\lambda_i(h_{e_{uv}}) = \mathbf{W}_\lambda h_{e_{uv}}; f(\vec{\mathbf{e}}_{uv}) = \vec{\mathbf{e}}_{uv}/||\vec{\mathbf{e}}_{uv}||. \tag{9}$$

We leave more elaborate choices of λ and f functions for future study.

Fictitious velocity integration. Similar to Satorras et al. (2021), we keep track of a fictitious velocity variable $\mathbf{v}_v$ for each node and linearly combine it (with weights $\mathbf{W}_\lambda$) update it and positions $\mathbf{x}_v$ in turns *like* a Euler-discretized Hamiltonian integration.

$$\mathbf{v}_v^{(k+1)} = \phi^{v\to\mathcal{V}}(h_v^{(k)})\mathbf{v}^{(k)} + \mathbf{W}_v \sum_i \sum_{u\in\mathcal{N}(v)} \lambda_i(h_{e_{uv}}^{(k)}) f(\vec{\mathbf{e}}_{uv}^k) \tag{10}$$

$$\mathbf{x}_v^{(k+1)} = \mathbf{x}_v^{(k)} + \mathbf{v}_v^{(k)} \tag{11}$$

During velocity update, we scale the current velocity and, for the sake of parameter saving, reuse the same set of linear combinations of edge vectors used in Equation 6 as the additive term.

Remark 3. *If $h_{e_{uv}}$ is distinct among edges, there exist sets of λ_i (even when $f = I$ is the identity) such that Equation 10 can approximate any vector on the subspace spanned by edge vectors and is thus universal up to equivariance thereon.*

Equivariance analysis. We inlay these modular components into the framework of graph neural networks (described in Section 2.2) to produce the SAKE architecture, which we show in Algorithm 1. (A $\mathrm{SAKEModel}$ is defined by applying $\mathrm{SAKELayer}$ iteratively from $k = 0$ to $k = K$, the depth of the network.) We have previously discussed that spatial attention is E(n)-invariant in Remark 4. With the E(n) equivariance of velocity update and position update (Equation 10) being proved in Satorras et al. (2021) and the rest of the model only takes the norm of edges and is E(n) invariant, SAKE is E(n)-equivariant.

Runtime analysis. The runtime complexity of spatial attention (Equation 6), which is the bottleneck of this algorithm, is $\mathcal{O}(|\mathcal{E}|N_\lambda CD)$, i.e., linear in the number of graph edges $|\mathcal{E}|$, number of attention weights N_λ, embedding dimension C, and geometric dimension D. When implemented, the FOR loops in Algorithm 1 can be packed into multi-dimensional tensors that could benefit from GPU acceleration.

In relation to spherical harmonics-based models. Under the framework of TFN (Thomas et al., 2018), the embedding and position input in Algorithm 1 $\{h_v^{(k)}\}, \{\mathbf{x}_v^{(k)}\}$ corresponds to the $l = 0$ and $l = 1$ type tensors. The concatenation followed by neural network in Equation 6 is loosely analogous to the direct sum operation in Clebsch-Gordon decomposition. While Smith et al. (2017); Schütt et al. (2017); Satorras et al. (2021) operates on $l = 0$ tensors only, the *spatial attention* mechanism we propose (Section 4 Equation 6) and velocity/position update (Section 5 Equation 10) corresponds to $1 \oplus 0 \to 0$ and $1 \oplus 0 \to 1$ type networks, respectively. Klicpera et al. (2021a) has discovered that $l = 0, 1$ are the complete levels of tensor to universally describe geometry on $E(3)$, while higher-order tensors are not necessary. Kovács et al. (2021) also discusses the concept of *density projection* where body-order functional forms can be recovered by lower-order terms.

6 EXPERIMENTS

As discussed in Section 2, SAKE provides a mapping from and to the joint space of geometry and embedding $\mathcal{X} \times \mathcal{H}$, while being equivariant on geometric space and invariant on embedding space. We are therefore interested to characterize the performance of SAKE on two types of tasks: *invariant modeling* (Section 6.1), where we model some scalar property of a physical system, namely potential energy; and *equivariant modeling* (Section 6.2), where we generate a (distribution of) coördinates conditioned on initial coördinates and embedding. See Appendix Section 9 for experimental details and settings.

6.1 INVARIANT TASKS: MACHINE LEARNING POTENTIAL CONSTRUCTION

MD17 (Chmiela et al., 2017) tests the capacity of the model in the extreme small data regime. Its training set contains merely 1000 configurations and quantum chemical energies and forces of single small molecules in vacuum computed using density functional theory (DFT). As summarized in Table 1, SAKE consistently outperforms all benchmarked models with the exception of NequIP (Batzner et al., 2021), which has comparable performance but is noticeably slower. In terms of training cost, most state-of-the-art models requires days of training, whereas the MD17 experiment was completed within 6 hours.

ISO17 (Schütt et al., 2017) goes beyond the single-molecule regime. It involves a slightly more diverse chemical space containing 5000-step *ab initio* molecular dynamics simulation trajectories (with energies and forces) of 129 molecules with the same formula $C_7H_{10}O_2$. The test set of ISO17 is split into *known* (which Kovács et al. (2021) argues to be very close to training set) and *unknown* molecules, based on the chemical identity (topology) at the beginning of the simulation, which could be regarded as interpolative and extrapolative tasks, respectively. As shown in Table 2, SAKE

Table 1: Inference time (ms) and test set energy (E) and force (F) mean absolute error (MAE) (meV and meV/Å) on the MD17 quantum chemical dataset.

Model		SchNet Schütt et al., 2017	DimeNet Klicpera et al., 2020b	sGDML Chmiela et al., 2019	PaiNN Schütt et al., 2021	GemNet(T/Q) Klicpera et al., 2021a	NequIP Batzner et al., 2021	SAKE
Inference time	batch of 32		65			88/376	206	**12**
	batch of 4		31			38/99	197	**4**
Aspirin	E	16.0	8.8	8.2	6.9	-	**5.3**	$6.46^{6.47}_{6.43}$
	F	58.5	21.6	29.5	14.7	9.4	**8.2**	$9.90^{9.92}_{9.88}$
Ethanol	E	3.5	2.8	3.0	2.7	-	**2.2**	$\mathbf{2.26}^{2.27}_{2.25}$
	F	16.9	10.0	14.3	9.7	3.7	3.8	$\mathbf{3.70}^{3.77}_{3.62}$
Malonaldehyde	E	5.6	4.5	4.3	3.9	-	**3.3**	$\mathbf{3.26}^{3.27}_{3.26}$
	F	28.6	16.6	17.8	13.8	6.7	5.8	$\mathbf{5.17}^{5.18}_{5.16}$
Naphtalene	E	6.9	5.3	5.2	5.0	-	**4.9**	$\mathbf{4.94}^{4.95}_{4.93}$
	F	25.2	9.3	4.8	3.3	2.2	**1.6**	$2.44^{2.44}_{2.44}$
Salicylic acid	E	8.7	5.8	5.2	4.9	-	**4.0**	$4.76^{4.77}_{4.75}$
	F	36.9	16.2	12.1	8.5	5.4	**3.9**	$5.14^{5.17}_{5.13}$
Toluene	E	5.2	4.4	4.3	4.1	-	**4.0**	$\mathbf{4.02}^{4.02}_{4.01}$
	F	24.7	9.4	6.1	4.1	2.6	**2.0**	$2.44^{2.44}_{2.43}$
Uracil	E	6.1	5.0	4.8	4.5	-	**4.5**	$\mathbf{4.52}^{4.54}_{4.50}$
	F	24.3	13.1	10.4	6.0	4.2	**3.3**	$4.05^{4.06}_{4.04}$

significantly outperforms other models on the unknown molecules in the test set, indicating that SAKE is capable of extrapolating onto unseen chemical spaces when trained on limited data.

QM9 (Ramakrishnan et al., 2014) tests the transductive generalizability across distinct small chemical graphs. It entails a very diverse chemical space of 134k molecules with annotated phsyical properties calculated with B3LYP/6-31G(2df,p) level of quantum chemistry, albeit all with at-equilibrium (low energy) conformations. In Table 3, SAKE achieves state-of-the-art performance at predicting HOMO, LUMO, and $\Delta\epsilon$ properties—a class of most crucial molecular properties closely related to reactivity. Interestingly, SAKE performs competitively on predicting *extensive* physical properties but not *intensive* ones (also see discussions in Pronobis et al. (2018)). We hypothesize that this will be mitigated by choosing size-invariant pooling functions—we leave this for future study.

Table 2: Test set energy (E) and force (F) mean absolute error (MAE) (meV and meV/Å) on known and unknown molecules in ISO17.

		ACE Kovács et al., 2021	SchNet Schütt et al., 2017	PhysNet Unke and Meuwly, 2019	SAKE
known	E	16	16	**4**	$12.17^{12.18}_{12.12}$
	F	43	43	**5**	$12.33^{12.34}_{12.31}$
unknown	E	85	104	127	$\mathbf{53.37}^{53.62}_{53.15}$
	F	85	95	60	$\mathbf{39.46}^{39.59}_{39.35}$

Table 3: QM9 test set performance.

	α Bohr^3	$\Delta\epsilon$ meV	HOMO meV	LUMO meV	μ D	C_v cal/mol K
SchNet Schütt et al., 2017	0.235	63	41	34	0.033	0.033
DimeNet++ Klicpera et al., 2020a	0.044	33	25	20	0.030	0.023
SE(3)-TF Fuchs et al., 2020	0.142	53	35	33	0.051	0.054
EGNN Satorras et al., 2021	0.071	48	29	25	0.029	0.031
PaiNN Schütt et al., 2021	0.059	36	46	20	**0.012**	**0.024**
TorchMD-Net Thölke and Fabritiis, 2022	0.059	36	20	17	**0.011**	**0.023**
SphereNet Liu et al., 2022	**0.030**	31	19	23	0.025	**0.022**
SAKE	0.068	**23**	**16**	**13**	**0.014**	0.087

6.2 EQUIVARIANT TASKS

Charged N-body dynamical system forecasting. (Kipf et al., 2018; Fuchs et al., 2020) tests if a model can predict the evolving of a physical system sufficiently long after initial conditions. It consists of 5 charge-carrying particles with initial positions and velocities, and the position at a given moment is predicted. As shown in Table 4, although the interactions (Coulomb forces) are entirely pairwise, we see here that the additional expressiveness SAKE affords lead to competitive performance on this demonstrative task.

Walking motion capture. (CMU, 2003) is used by us to demonstrate SAKE's general capacity to forecast dynamic systems beyond microscopic scale. In this task, closely following the experiment setting of Huang et al. (2022); Kipf et al. (2018), we predict the position of a walking person

(subject 35 in CMU motion capture database (CMU, 2003)) based on their initial position. Again, we observe that SAKE outperform other models by a large margin (Table 5) and is significantly faster.

Table 4: Mean Squared Error (MSE) and inference time (ms)for dynamic system forecasting.

Architecture	MSE	Inference time
SE(3)-TF (Fuchs et al., 2020)	0.244	0.1346
TFN (Thomas et al., 2018)	0.155	0.0343
GNN (Kipf and Welling, 2016)	0.0107	0.0032
EGNN (Satorras et al., 2021)	0.0071	0.0062
SAKE	0.0049	0.0079
SEGNN (Brandstetter et al., 2021)	**0.0043**	0.0260

6.3 EQUIVARIANT EXACT LIKELIHOOD SAMPLING WITH SAKE FLOW.

Current equivariant normalizing flow models (Satorras et al., 2022; Köhler et al., 2020) (brielfy reviewed in Appendix Section 10), relies on ODE-based numerical integration(Chen et al., 2019), are computationally expensive. We propose a much simpler invertible flow model that uses our SAKE model, termed SAKE Flow. First, following a scheme introduced in Huang et al. (2020) (in which it is argued that with flexible enough kernels Equation 23 could be approximated arbitrarily well), we extend the space $\mathcal{X}$ with an auxiliary space $\mathcal{A}$. Correspondingly, we extend the tractable distribution to be $q(\mathbf{z}, \mathbf{a}) = q_z(\mathbf{z})q_a(\mathbf{a})$ and the target distribution to be $p(\mathbf{x}, \mathbf{a}) = p_x(\mathbf{z})q_a(\mathbf{a})$. We then change the problem statement of Equation 22 to: find a parametrized function $f_{zx}(\mathbf{z}, \mathbf{a}; \theta) = f_{xz}^{-1}(\mathbf{x}, \mathbf{a}; \theta)$ that is a bijection on the space $\mathcal{A} \times \mathcal{X}$ to maximize the joint likelihood:

$$\hat{\theta} = \operatorname{argmax} \mathbb{E}_{\mathbf{a} \sim q_a}[\log p(\mathbf{x}, \mathbf{a}))] \tag{12}$$

which is, up to a constant, an evidence lower bound for $\mathbf{x}$:

$$\log p_x(\mathbf{x}) \tag{13}$$
$$= \mathbb{E}_{q_a}[\log p(\mathbf{x}, \mathbf{a})] + D_{\mathrm{KL}}[q(\mathbf{a})||p(\mathbf{a}|\mathbf{x})] + H_a \tag{14}$$
$$\geq \mathbb{E}_{q_a}[\log p(\mathbf{x}, \mathbf{a})] + H_a, \tag{15}$$

The equality holds when the (non-negative) Kullback–Leibler divergence between the tractable distribution q_a and the conditional distribution given samples $p(\mathbf{a}|\mathbf{x})$ is zero. As such, an unbiased estimate of the marginal likelihood can be given by

$$\log \hat{p}(\mathbf{x}) = \log p(\mathbf{x}, \mathbf{a}) - \log q_a(\mathbf{a}). \tag{16}$$

Similar to Huang et al. (2020); Dinh et al. (2017), we define $f_{zx}(\mathbf{z}, \mathbf{a}, \theta)$ as a series of alternating affine coupling:

$$g^{z \to a/a \to z}(\mathbf{z}, \mathbf{a}) = \mathbf{z}, \exp(S^{z \to a/a \to z}(\mathbf{z})) \odot \mathbf{a} + T^{z \to a/a \to z)}, \tag{17}$$

with analytic inverse. Composing these transformations $g^{z \to a} \circ g^{a \to z} \circ ...g^{z \to a} \circ g^{a \to z}$, we get our bijection $f_{zx}(\mathbf{z}, \mathbf{a}; \theta)$.

Table 5: Walking motion capture performance.

	GNN	EGNN (Satorras et al., 2021)	GMN (Huang et al., 2022)	SAKE
MAE	67.3±1.1	59.1±2.1	43.9±1.1	**22.7** ±1.6
Epoch time			5.66 s	**1.81** s

Now, it only remains to define the structure of the translation functions $\{T\}$ and scaling functions $\{S\}$. As is outlined in Satorras et al. (2022), if f_{zx} is *translation-invariant*, it is impossible to have $\int p_x(\mathbf{x})\, \mathrm{d}\, x = 1$, so we drop the translation invaraince requirement and design f_{zx}, as well as the composing $\{T, S\}$, to be only rotation and reflection equivariant. Correspondingly, we require all tractable distributions to be confined on a $(|\mathcal{V}| - 1)n$-dimensional subspace with $\mathbf{0}$ gravity center. (Note that this reduce the symmetry group we work on from E(n) to SO(n).) A valid choice both for $q_z(\mathbf{z})$ and $q_a(\mathbf{a})$ with is a *centered Gaussian* distribution (Satorras et al., 2022):

$$p(\mathbf{z}) = \frac{1}{(2\pi)^{|\mathcal{V}-1|n/2}} \exp(-\frac{1}{2}||\mathbf{z}||^2) \tag{18}$$

with $\sum_{v \in \mathcal{V}} \mathbf{x}_v = \mathbf{0}$. Moreover, to keep the gravity center at $\mathbf{0}$, we require that $\{T, S\}$ shall not change the gravity center. A general recipe to construct such $\{T, S\}$ is to use the *equivariant* and *invariant* outputs of SAKE to parametrize T and S respectively.

$$\begin{gathered} h, \mathbf{x} = \mathrm{SAKEModel}(h, \mathbf{x}); \mathbf{x} = \mathbf{x} - \mathrm{MEAN}(\mathbf{x}); h = \exp(\mathrm{MEAN}(h)); \\ T(\mathbf{z}) = \mathbf{z} + \mathbf{x}; S(\mathbf{z}) = h * \mathbf{z} \end{gathered} \tag{19}$$

To preserve the center of gravity, we center the translation to have zero center of gravity and enforce the same scalar to be used across particles as the scaling factor.

To experimentally benchmark this innovation, we adopted the benchmark systems of DW4 and LJ13 in Köhler et al. (2020), with 4 particles on 2-dimensional space and 13 particles on 3-dimensional space, whose energy landscapes are respectively determined by a pairwise double-well potential and a Lennard-Jones potential. As shown in Table 6, SAKE consistently outperforms other models with the exception of DW4 system with 10^3 training examples. The slightly worse performance on this particular case might be attributed to the fact that SAKEFlow discretizes (with stochasticity) the integration of Equation 23 and requires one set of parameters for each layer rather than shared parameters as in Köhler et al. (2020); Satorras et al. (2022). The overall superior performance SAKE demonstrated in this task is indicative that SAKE can be used in the direct sampling of high-dimensional many-body conformations, namely Boltzmann type distributions $p(\mathbf{x}) \propto \exp(-u(\mathbf{x}))$ as discussed in Noé et al. (2019)—we leave this for future study.

Table 6: Test set negative log likelihood (NLL) with different generative method and amount of training data.

	DW4			LJ13		
# Training samples	10^2	10^3	10^4	10^2	10^3	10^4
Graph Normalizing Flow (Liu et al., 2019)	11.31	10.38	7.95	42.84	37.17	36.49
Simple dynamics (Köhler et al., 2020)	9.51	9.53	9.47	33.10	32.79	32.99
Kernel Dynamics (Köhler et al., 2020)	8.74	8.67	8.42	31.49	31.17	31.25
Equivariant Normalizing Flow (Satorras et al., 2022)	**8.31**	8.15	7.69	30.99	30.56	30.41
SAKE	9.70	**7.78**	**3.85**	**25.21**	**23.29**	**22.42**

7 Discussion

Conclusions In this paper we have introduced an invariant/equivariant functional form termed *spatial attention* that uses neurally parametrized linear combinations of edge vectors to equivariantly yet universally characterize node environments at a fraction of the cost of more expensive equivariant approaches that makes use of spherical harmonics. Equipped with this spatial attention module, we use it to build a *spatial attention kinetic network* (SAKE) architecture which is permutationally, translationally, rotationally, and reflectionally equivariant. We have demonstrated the utility of this model in n-body physical modeling tasks ranging from potential energy prediction to dynamical system forecasting and, when equipped with an augmented invertible flow scheme, sampling.

Limitations In this work, apart from the novel functional form spatial attention (Equation 6), the rest of the architecture has not been thoroughly optimized and analyzed in the context of the growing design space of equivariant neural networks—an ablation study might be needed to clearly distinguish which components of the network are contributing to the high performance the most. For example, thorough experiments are needed to study the exact likelihood sampling performance of $\mathrm{SAKEModel}$ with ODE integration or *vise versa*—other equivariant model with noise-augmented generative process.

Future directions We plan to conduct more thorough experiments on (bio)molecular systems to explore the potential of SAKE in building general protein/small molecule force fields and enhanced sampling methods that could facilitate large-scale simulations useful in therapeutics and material discovery.

Social impact and ethics statement This work provide an accurate and extremely efficient way to approximate properties and dynamics of molecular systems and physical states. It may advances research in a wide range of disciplines, including physics, chemistry, biochemistry, biophysics, and drug and material discovery. As with all molecular machine learning methods, negative implications may be possible if used in the design of explosives, toxins, chemical weapons, and overly addictive recreational narcotics.

Reproducibility statement The software package containing the algorithm proposed here is distributed open source under MIT license. All necessary code, data, and details to reproduce the experiments can be found in Appendix Section 9.

REFERENCES

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016. URL http://arxiv.org/abs/1609.02907.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. *arXiv preprint arXiv:1704.01212*, 2017.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, pages 1024–1034, 2017.

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E(n) equivariant graph neural networks. *CoRR*, abs/2102.09844, 2021. URL https://arxiv.org/abs/2102.09844.

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.

Nathaniel Thomas, Tess Smidt, Steven M. Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation- and translation-equivariant neural networks for 3d point clouds. *CoRR*, abs/1802.08219, 2018. URL http://arxiv.org/abs/1802.08219.

Johannes Klicpera, Florian Becker, and Stephan Günnemann. Gemnet: Universal directional graph neural networks for molecules. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021a.

Fabian B. Fuchs, Daniel E. Worrall, Volker Fischer, and Max Welling. Se(3)-transformers: 3d roto-translation equivariant attention networks. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, 2020.

Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P. Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E. Smidt, and Boris Kozinsky. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials, 2021.

Brandon Anderson, Truong-Son Hy, and Risi Kondor. Cormorant: Covariant molecular neural networks, 2019.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *CoRR*, abs/1901.00596, 2019. URL http://arxiv.org/abs/1901.00596.

Jiaqi Han, Yu Rong, Tingyang Xu, and Wenbing Huang. Geometrically equivariant graph neural networks: A survey, 2022. URL https://arxiv.org/abs/2202.07230.

J. S. Smith, O. Isayev, and A. E. Roitberg. Ani-1: an extensible neural network potential with dft accuracy at force field computational cost. *Chemical Science*, 8(4):3192–3203, 2017. ISSN 2041-6539. doi: 10.1039/c6sc05720a. URL http://dx.doi.org/10.1039/C6SC05720A.

Jörg Behler. Atom-centered symmetry functions for constructing high-dimensional neural network potentials. *The Journal of Chemical Physics*, 134(7):074106, 2011. doi: 10.1063/1.3553717. URL https://doi.org/10.1063/1.3553717.

Johannes Klicpera, Shankari Giri, Johannes T. Margraf, and Stephan Günnemann. Fast and uncertainty-aware directional message passing for non-equilibrium molecules. *CoRR*, abs/2011.14115, 2020a. URL https://arxiv.org/abs/2011.14115.

Yi Liu, Limei Wang, Meng Liu, Yuchao Lin, Xuan Zhang, Bora Oztekin, and Shuiwang Ji. Spherical message passing for 3d molecular graphs. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=givsRXsOt9r.

Johannes Klicpera, Chandan Yeshwanth, and Stephan Günnemann. Directional message passing on molecular graphs via synthetic coordinates. *CoRR*, abs/2111.04718, 2021b. URL https://arxiv.org/abs/2111.04718.

Johannes Brandstetter, Rob Hesselink, Elise van der Pol, Erik Bekkers, and Max Welling. Geometric and physical quantities improve E(3) equivariant message passing. *CoRR*, abs/2110.02905, 2021. URL https://arxiv.org/abs/2110.02905.

Kristof T. Schütt, Oliver T. Unke, and Michael Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra, 2021.

Philipp Thölke and Gianni De Fabritiis. Torchmd-net: Equivariant transformers for neural network based molecular potentials. *CoRR*, abs/2202.02541, 2022. URL https://arxiv.org/abs/2202.02541.

Wenbing Huang, Jiaqi Han, Yu Rong, Tingyang Xu, Fuchun Sun, and Junzhou Huang. Equivariant graph mechanics networks with constraints. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=SHbhHHfePhP.

Chin-Wei Huang, Laurent Dinh, and Aaron Courville. Augmented normalizing flows: Bridging the gap between generative flows and latent variable models, 2020.

Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael J. L. Townshend, and Ron Dror. Learning from protein structure with geometric vector perceptrons, 2021.

Vijay Prakash Dwivedi, Chaitanya K. Joshi, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *CoRR*, abs/2003.00982, 2020. URL https://arxiv.org/abs/2003.00982.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, 2018.

Jonathan T. Barron. Continuously differentiable exponential linear units, 2017. URL https://arxiv.org/abs/1704.07483.

Albert Matveev, Alexey Artemov, Denis Zorin, and Evgeny Burnaev. Geometric attention for prediction of differential properties in 3d point clouds, 2020.

Dávid Péter Kovács, Cas van der Oord, Jiri Kucera, Alice EA Allen, Daniel J Cole, Christoph Ortner, and Gábor Csányi. Linear atomic cluster expansion force fields for organic molecules: beyond rmse. *Journal of chemical theory and computation*, 17(12):7696–7711, 2021.

Johannes Klicpera, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. *CoRR*, abs/2003.03123, 2020b. URL https://arxiv.org/abs/2003.03123.

Stefan Chmiela, Huziel E. Sauceda, Igor Poltavsky, Klaus-Robert Müller, and Alexandre Tkatchenko. sGDML: Constructing accurate and data efficient molecular force fields using machine learning. *Computer Physics Communications*, 240:38–45, jul 2019. doi: 10.1016/j.cpc.2019.02.007. URL https://doi.org/10.1016%2Fj.cpc.2019.02.007.

Stefan Chmiela, Alexandre Tkatchenko, Huziel E. Sauceda, Igor Poltavsky, Kristof T. Schütt, and Klaus-Robert Müller. Machine learning of accurate energy-conserving molecular force fields. *Science Advances*, 3(5), May 2017. ISSN 2375-2548. doi: 10.1126/sciadv.1603015. URL http://dx.doi.org/10.1126/sciadv.1603015.

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.

Wiktor Pronobis, Kristof T Schütt, Alexandre Tkatchenko, and Klaus-Robert Müller. Capturing intensive and extensive dft/tddft molecular properties with machine learning. *The European Physical Journal B*, 91(8):1–6, 2018.

Oliver T. Unke and Markus Meuwly. PhysNet: A neural network for predicting energies, forces, dipole moments, and partial charges. *Journal of Chemical Theory and Computation*, 15(6): 3678–3693, may 2019. doi: 10.1021/acs.jctc.9b00181. URL https://doi.org/10.1021%2Facs.jctc.9b00181.

Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. Neural relational inference for interacting systems, 2018. URL https://arxiv.org/abs/1802.04687.

CMU. Cmu motion capture database. http://mocap.cs.cmu.edu, 2003.

Victor Garcia Satorras, Emiel Hoogeboom, Fabian B. Fuchs, Ingmar Posner, and Max Welling. E(n) equivariant normalizing flows, 2022.

Jonas Köhler, Leon Klein, and Frank Noé. Equivariant flows: Exact likelihood generative learning for symmetric densities, 2020.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations, 2019.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp, 2017.

Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457):eaaw1147, 2019. doi: 10.1126/science.aaw1147. URL https://www.science.org/doi/abs/10.1126/science.aaw1147.

Jenny Liu, Aviral Kumar, Jimmy Ba, Jamie Kiros, and Kevin Swersky. Graph normalizing flows. *CoRR*, abs/1905.13177, 2019. URL http://arxiv.org/abs/1905.13177.

Timothy F Havel. Distance geometry: Theory, algorithms, and chemical applications. *Encyclopedia of Computational Chemistry*, 120:723–742, 1998.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989. ISSN 0893-6080. doi: https://doi.org/10.1016/0893-6080(89)90020-8. URL https://www.sciencedirect.com/science/article/pii/0893608089900208.

Roberto Bolaño. *2666*. Anagrama, Barcelona, 2004.

Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows, 2016.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference, 2021.

Danilo Jimenez Rezende, Sébastien Racanière, Irina Higgins, and Peter Toth. Equivariant hamiltonian flows, 2019. URL https://arxiv.org/abs/1909.13739.

Changyou Chen, Chunyuan Li, Liqun Chen, Wenlin Wang, Yunchen Pu, and Lawrence Carin. Continuous-time flows for efficient inference and density estimation, 2018.

8 PROOFS

8.1 PROOF FOR THEOREM 1

Proof. (Informal) An appropriate set of choices to satisfy the requirement is:

$f = I$, and λ takes the set of $|\mathcal{N}(v)| + |\mathcal{N}(v)|^2$ elements:

$$\left\{\lambda_i(h_{e_{u_j v}}) = [i = j]\right\} \cup \left\{\lambda_{kl}(h_{e_{u_j v}}) = [k = j] - [l = j]\right\} \tag{20}$$

where $[\cdot]$ is the Iverson bracket. The first $|\mathcal{N}(v)|$-set of one-indexed functions give the node-to-neighbor distances whereas the second $|\mathcal{N}(v)|^2$-set of two-indexed functions give the distances among neighbors. This recovers the distance matrix $A_{\mathcal{X}}(v)$ among the node and its neighbors $A_{\mathcal{X}}(v)_{ii} = ||\mathbf{x}_v - \mathbf{x}_{u_i}||$ and $A_{\mathcal{X}}(v)_{ij} = ||\mathbf{x}_{u_j} - \mathbf{x}_{u_i}||, i \neq j, 1 \leq i, j \leq |\mathcal{N}(v)|$. As such, the relative positions of the node and its neighbors are uniquely defined up to E(n) symmetry (Havel, 1998). In other words, $\mathbf{x}_v$ and $\mathbf{x}_{u_i}$ can be uniquely embedded on an arbitrary coördinate system on which g is evaluated. Since $\{\lambda\}$ is neural, by the universal approximation theorem (Hornik et al., 1989), $A_{\mathcal{X}}(v)$ can still be recovered, at least through recovering the set of $\{\lambda\}$ discussed above. Considering again the universal approximation of μ, any function of the geometric node environment can be approximated by ϕ^{SA}. □

8.2 PROOF FOR REMARK 4

Proof. We set $\mu = I$ without loss of generality. *Necessity:* given a parametrized dot product scalarization $\phi^{\mathrm{DPS}}(v)$ with weight-generating functions $\lambda_{q,i}(h_{e_{uv}}), \lambda_{k,i}(h_{e_{uv}})$ and edge equivariant functions f, for spatial attention, choose $\lambda_+ = \{\lambda_{u,i} + \lambda_{k,i}\}$ and $\lambda_- = \{\lambda_{u,i} - \lambda_{k,i}\}$ with the same f. As such $\phi^{\mathrm{DPS}}(v; \lambda_k, \lambda_q) = \frac{1}{2}\phi^{\mathrm{SA}}(v; \lambda_+)^2 - \frac{1}{2}\phi^{\mathrm{SA}}(v; \lambda_-)^2$. Hence $\phi^{\mathrm{SA}}(v; \lambda_+ \cup \lambda_-)$ fully recovers $\phi^{\mathrm{DPS}}(v; \lambda_k, \lambda_q)$. Naturally if $\phi^{\mathrm{DPS}}(v_0; \lambda_k, \lambda_q) \neq \phi^{\mathrm{DPS}}(v_1; \lambda_k, \lambda_q)$ we have $\phi^{\mathrm{SA}}(v; \lambda_+ \cup \lambda_-)(v_0) \neq \phi^{\mathrm{SA}}(v; \lambda_+ \cup \lambda_-)(v_1)$. *Insufficiency:* Consider nodes v_0 and v_1 with all orthogonal edges of different lengths. Then $\phi^{\mathrm{DPS}}(v_0) = \phi^{\mathrm{DPS}}(v_1) = 0$ always. For spatial attention, choose $\lambda_i = f = I$, edge lengths are recovered, $\phi^{\mathrm{SA}}(v_0) \neq \phi^{\mathrm{SA}}(v_1)$. □

9 DETAILED METHODS

9.1 CODE AVAILABILITY

The corresponding software package and all scripts used to conduct the experiments in this paper are distributed open source under MIT license at: `https://anonymous.4open.science/r/sake-EF30`

9.2 HARDWARE CONFIGURATION

All models are trained on NVIDIA Tesla V100 GPUs. Following the settings reported in the publications of baseline models, the inference time benchmark experiments (Section 6.2, 6.1) are done on NVIDIA GeForce GTX 1080 Ti GPU (For Table 1 and Table 5) and NVIDIA GeForce GTX 2080 Ti GPU (For Table 4).

9.3 ARCHITECTURE AND OPTIMIZATION DETAILS

One-layer feed-forward neural networks are used as f_r in Equation 2 (edge update); two-layer feed-forward neural networks are used as ϕ^e in Equation 2 (edge update), ϕ^v in Equation 4 (node update), $\phi^{v\to\mathcal{V}}$ in Equation 10 (velocity update), and μ in Equation 6 (spatial attention). SiLU is used everywhere as activation, except in Equation 10 (velocity update) where the last activation function is chosen as $y = 2 * \mathrm{Sigmoid}(x)$ to constraint the velocity scaling to between 0 and 2 and in Equation 8 where CeLU is used before attention; additionally, $\tanh$ is applied on the additive part of Equation 10 to constraint it to between -1 and 1. 4 attention heads are used with γ in Equation 8 spaced evenly between 0 and 5 Å. 50 RBF basis are used, spacedly evenly between 0 and 5 Å. All models are optimized with Adam optimizer. We summarize the hyperparameters used in these experiments in Table 7. All random seeds are fixed as 2666, in an homage to Bolaño (2004).

Experiment	Depth	Width	Learning Rate	Epochs	Batch Size	L2 Regularization
MD17 (Table 1)	6	64	10^{-3*}	5000	4	10^{-5}
ISO17 (Table 2)	6	64	10^{-3*}	100	128	10^{-12}
QM9 (Table 3)	6	32	10^{-4*}	5000	32	10^{-10}
N-Body Forecast (Table 4)	4	32	$5 * 10^{-4}$	1000	100	10^{-12}
DW4 (Table 6)	4	64	10^{-5}	2000	100	10^{-5}
LJ13 (Table 6)	4	64	10^{-5}	1000	100	10^{-5}

Table 7: Hyperparameters used in experiments (* A cosine warm up and annealing schedule is used, where the learning rate is gradually increased from 10^{-6} to the peak value in the first 10% epochs and decreased in the rest 90%.)

9.4 Data Availability

The source and details of benchmark datasets are summarized in Table.

Experiment	License	Size	Split
MD17 http://quantum-machine.org/gdml/#datasets (Table 1)		8 Systems; 100K-1M snapshots	Random: 1K Train
ISO17 http://quantum-machine.org/datasets/(Table 2)		129 Molecules; 5000 snapshots	Fixed
QM9 http://quantum-machine.org/datasets/		135k molecules	Fixed
N-Body Forecast [1](Table 4)	MIT	5 particles	Fixed: 3K Train; 2K Valid; 2K Test
DW4 https://github.com/vgsatorras/en_flows(Table 6)	MIT	4 particles	Random
LJ13 https://github.com/vgsatorras/en_flows(Table 6)	MIT	13 particles	Random

Table 8: Dataset details.

10 Brief introduction of equivariant normalizing flows

Normalizing flows (Rezende and Mohamed, 2016; Papamakarios et al., 2021) are a family of learnable bijections $f_{zx} : \mathcal{Z} \to \mathcal{X}$ that transform a tractable distribution on latent space $q_z(\mathbf{z})$ to another on the target space $\mathbf{x} = f_{zx}(\mathbf{z}; \theta)$ (dropping dependency on parameter thereafter) whose density can be analytically written as

$$\log q_x(\mathbf{x}) = \log q_z(\mathbf{z}) + \log \det |\frac{\partial f_{xz}(\mathbf{x})}{\partial \mathbf{x}}| \tag{21}$$

and conversely from a sample on the target space to the latent space $\mathbf{z} = f_{xz}(\mathbf{x}; \theta)$ whose likelihood is given by

$$\log p_z(\mathbf{z}) = \log p_x(\mathbf{x}) + \log \det |\frac{\partial f_{zx}(\mathbf{z})}{\partial \mathbf{z}}| \tag{22}$$

where $f_{zx} = f_{xz}^{-1}$. To close the gap between the intractable p_x and the tractable q_x, so that one can sample on the target space efficiently, the flow is then trained by maximizing either Equation 21, if an unnormalized target distribution is given, or Equation 22, if samples are given.

The concept of *equivariant* normalizing flow was first introduced in Köhler et al. (2020); Rezende et al. (2019); Satorras et al. (2022), where they adopted the framework of continuous normalizing flow (Chen et al., 2018) to define the bijection f_{zx} as

$$\mathbf{x} = \int_0^1 f'_{zx}(\mathbf{z}(t)) \, \mathrm{d}\, t \tag{23}$$

and restrict f'_{zx} as E(n)-equivariant w.r.t. $\mathbf{z}$. Integration, or the sum over infinitely many equivariant functions, does not alter the equivariance. To numerically approximate this integration Chen et al. (2019) involves evaluating f'_{zx} multiple times and is therefore expensive.

11 Ablation study

To elucidate the contribution of each attention modules on the performance of SAKE, we delete certain parts of it and retrain the MD17 force prediction task. Interestingly, depriving the model of either semantic or Euclidean attention alone doesn't drastically change it performance, while deleting both is detrimental, possibly suggesting these two attention mechanisms complement each other.

	Aspirin	Ethanol	Malonaldehyde	Naphathene	Salicylic Acid	Toluene	Uracil
SAKE	9.90	3.70	5.17	2.44	5.14	2.44	4.05
No spatial attention	424.67	93.29	382.68	239.94	436.15	747.30	212.59
No semantic attention	9.17	3.49	7.28	2.68	5.59	2.54	4.09
No Euclidean attention	9.62	3.81	5.34	2.44	5.24	2.51	3.95
No semantic or Euclidean attention	10.39	13.35	14.88	9.88	9.56	44.73	4.09
No update	11.68	3.52	5.79	2.55	5.32	3.25	4.05
EGNN	298.05	169.96	245.82	138.36	199.07	377.73	165.77
EGNN + RBF	68.39	41.11	77.40	36.08	43.41	48.84	42.59
EGNN + semantic attention	123.45	95.11	124.05	114.83	131.15	91.63	85.95
EGNN + RBF + semantic attention	58.10	43.65	72.96	40.65	42.83	29.88	39.01