

A PRELIMINARIES

A.1 ISOMETRIES

Definition A.1. Let (X, d_X) and (Y, d_Y) be metric spaces with metrics $d_X : X \times X \rightarrow \mathbb{R}$, $d_Y : Y \times Y \rightarrow \mathbb{R}$. An isometry $\varphi : X \rightarrow Y$ is a distance-preserving isomorphism if

$$d_X(a, b) = d_Y(\varphi(a), \varphi(b)) \quad (15)$$

for all $a, b \in X$.

A.2 INVARIANCE & EQUIVARIANCE

Definition A.2. Let $\rho_V : G \rightarrow GL(K, V)$ be a representation of group G , and let $\rho_V(g) : V \xrightarrow{\sim} V$ be an automorphism on V for $g \in G$. A function $f : V \rightarrow W$ is G -equivariant if there exists an equivalent representation $\rho_W : G \rightarrow GL(K, W)$ with equivalent automorphism $\rho_W(g) : W \xrightarrow{\sim} W$, such that

$$f(\rho_V(g)(v)) = \rho_W(g)(f(v)) \quad (16)$$

for all $v \in V$ and $g \in G$.

Definition A.3. Let $\rho : G \rightarrow GL(K, V)$ be a representation of group G and let $\rho(g) : V \xrightarrow{\sim} V$ be an automorphism for $g \in G$. A function $f : V \rightarrow W$ is G -invariant if

$$f(\rho(g)(v)) = f(v) \quad (17)$$

for all $v \in V$ and $g \in G$.

A.3 SPECIAL PROPERTY OF $SO(n)$

Since rotations are distance, angle, and orientation preserving, they are linear transformations. As such, rotations can be represented as a matrix. Suppose $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ and $Q \in \mathbb{R}^{n \times n}$ is a rotation matrix. Then if Q is an isometry, we require: $\mathbf{x}^\top \mathbf{y} = (Q\mathbf{x})^\top (Q\mathbf{y}) = \mathbf{x}^\top (Q^\top Q)\mathbf{y}$ or $Q^\top Q = I = Q^\top Q$. Preservation of orientation (equivalently, handedness) means $\det Q > 0$. We take the determinant of the identity and find $\det(Q^\top Q) = (\det(Q))^2 = 1$, which means $\det Q = \pm 1$. Hence, $\det Q = +1$ for $SO(n)$.

A.4 PROOF OF $E(n)$ -INVARIANCE FOR SCHNET CONTINUOUS FILTER

In SchNet, the representation of interactions of a particle i in the next layer $l + 1$ is given by the convolution with neighboring particles:

$$\mathbf{h}_i^{(l+1)} := (H * W) = \sum_{j=1}^N \mathbf{h}_j^{(l)} \circ W_\theta \begin{bmatrix} e_1(\mathbf{x}_j - \mathbf{x}_i) \\ \vdots \\ e_n(\mathbf{x}_j - \mathbf{x}_i) \end{bmatrix} \quad (18)$$

where $e_k(\mathbf{x}_j - \mathbf{x}_i) = \exp(-\gamma(\|\mathbf{x}_j - \mathbf{x}_i\|_2 - \mu_k)^2)$.

Proof. Consider an isometry of $E(n)$ defined by $T : \mathbf{x} \mapsto A\mathbf{x} + \mathbf{b}$ for $A \in \mathbb{R}^{n \times n}$ an orthogonal matrix rotation and $\mathbf{b} \in \mathbb{R}^n$ a translation vector. Then the basis expansion becomes:

$$\begin{aligned} e_k(T(\mathbf{x}_j) - T(\mathbf{x}_i)) &= \exp(-\gamma(\|T(\mathbf{x}_j) - T(\mathbf{x}_i)\|_2 - \mu_k)^2) \\ &= \exp(-\gamma(\|A\mathbf{x}_j + \mathbf{b} - A\mathbf{x}_i - \mathbf{b}\|_2 - \mu_k)^2) \\ &= \exp(-\gamma((\mathbf{x}_j - \mathbf{x}_i)^\top A^\top A(\mathbf{x}_j - \mathbf{x}_i) - \mu_k)^2) \\ &= \exp(-\gamma(\|\mathbf{x}_j - \mathbf{x}_i\|_2 - \mu_k)^2). \end{aligned} \quad (19)$$

It follows that the basis expansion is $E(n)$ -invariant and, thus, so is the continuous convolution. \square

B PROOF OF EQUIVARIANCES FOR TEMPORAL ATTENTION LAYER

B.1 POSITION COMPONENT: $E(n)$ -EQUIVARIANCE

Proof. Consider an isometry of $E(n)$ defined by $\xi \mapsto A\xi + \mathbf{b}$ for $A \in \mathbb{R}^{n \times n}$ an orthogonal rotation matrix and $\mathbf{b} \in \mathbb{R}^n$ a translation vector. Then

$$\begin{aligned} \|\xi_i(t) - \xi_i(s)\|^2 &= \|A\xi_i(t) + \mathbf{b} - A\xi_i(s) - \mathbf{b}\|^2 \\ &= \|A(\xi_i(t) - \xi_i(s))\|^2 \\ &= (\xi_i(t) - \xi_i(s))^\top A^\top A (\xi_i(t) - \xi_i(s)) \\ &= \|\xi_i(t) - \xi_i(s)\|^2, \end{aligned} \tag{20}$$

where the last line follows by the orthogonality of A . Hence, $\Psi_e(\theta_i(t), \theta_i(s), \|\xi(t) - \xi(s)\|_2^2)$ is $E(n)$ -invariant. Thus, applying the isometry to the layer

$$\tilde{\xi}_i(t) = \xi_i(t) + \sum_{\substack{s=1 \\ s \neq t}}^L (\xi_i(t) - \xi_i(s)) \phi_{\text{inf}}(\mathbf{m}_i(t, s)) \tag{21}$$

yields

$$\begin{aligned} &A\xi_i(t) + \mathbf{b} + \sum_{\substack{s=1 \\ s \neq t}}^L A(\xi_i(t) - \xi_i(s)) \phi_{\text{inf}}(\mathbf{m}_i(t, s)) \\ &= A \left(\xi_i(t) + \sum_{\substack{s=1 \\ s \neq t}}^L (\xi_i(t) - \xi_i(s)) \phi_{\text{inf}}(\mathbf{m}_i(t, s)) \right) + \mathbf{b} \\ &= A\tilde{\xi}_i(t) + \mathbf{b}. \end{aligned} \tag{22}$$

It follows that the position component of ETAL is $E(n)$ -equivariant. \square

B.2 VELOCITY COMPONENT: $SO(n)$ -EQUIVARIANCE

Proof. Consider an orthogonal rotation matrix $A \in \mathbb{R}^{n \times n}$ of $SO(n)$ with action $\omega \mapsto A\omega$. Clearly,

$$\begin{aligned} (A\omega_i(t))^\top (A\omega_i(s)) &= \omega_i(t)^\top A^\top A \omega_i(s) \\ &= \omega_i(t)^\top \omega_i(s). \end{aligned} \tag{23}$$

Thus, the coefficient $\beta_i(t, s) = \omega_i(t)^\top \omega_i(s) / \sum_{s'=1}^L \exp(\omega_i(t)^\top \omega_i(s'))$ is invariant under rotations of $SO(n)$.

Applying the rotation to the attention layer

$$\tilde{\omega}_i(t) = \sum_{s=1}^L \beta_i(t, s) \omega_i(s) \tag{24}$$

yields:

$$\begin{aligned} \sum_{s=1}^L \beta_i(t, s) A\omega_i(s) &= A \sum_{s=1}^L \beta_i(t, s) \omega_i(s) \\ &= A\tilde{\omega}_i(t), \end{aligned} \tag{25}$$

as asserted. Hence, the velocity component of ETAL is $SO(n)$ -equivariant. \square

C TENSORIZATION OF EQUIVARIANT TEMPORAL ATTENTION LAYER (ETAL)

C.1 TEMPORAL ATTENTION FEATURE COMPONENT

Define the matrix of spatial graph representations for features across time:

$$\theta_i^{[1:L]} = \begin{bmatrix} \theta_i(1)^\top \\ \vdots \\ \theta_i(L)^\top \end{bmatrix} \in \mathbb{R}^{L \times d}. \quad (26)$$

We compute the value vectors $v_i^{[1:L]} = \theta_i^{[1:L]} V \in \mathbb{R}^{L \times d}$, key vectors $k_i^{[1:L]} = \theta_i^{[1:L]} K \in \mathbb{R}^{L \times d}$, and query vectors $q_i^{[1:L]} = \theta_i^{[1:L]} Q \in \mathbb{R}^{L \times d}$ for $V, K, Q \in \mathbb{R}^{d \times d}$. Thus, to compute the attention layer for the graph features in Equation 7, we first compute the weights:

$$\alpha_i = \text{softmax} \left(\frac{\theta_i^{[1:L]} Q K^\top \theta_i^{[1:L]^\top}}{\sqrt{d}} \right) \in \mathbb{R}^{L \times L}. \quad (27)$$

Then we apply the weights as follows:

$$\tilde{\theta}_i^{[1:L]} = \alpha_i v_i^{[1:L]} \in \mathbb{R}^{L \times d}. \quad (28)$$

We would like to tensorize this computation for all nodes $i = 1, \dots, N$. Let $\theta^{[1:L]} = (\theta_1^{[1:L]}, \dots, \theta_N^{[1:L]}) \in \mathbb{R}^{N \times L \times d}$. Likewise, let $k^{[1:L]} = \theta^{[1:L]} K \in \mathbb{R}^{N \times L \times d}$, $q^{[1:L]} = \theta^{[1:L]} Q \in \mathbb{R}^{N \times L \times d}$, and $v^{[1:L]} = \theta^{[1:L]} V \in \mathbb{R}^{N \times L \times d}$. In a slight abuse of notation,

$$\alpha = \text{softmax} \left(\frac{q^{[1:L]} k^{[1:L]^\top}}{\sqrt{d}} \right) \in \mathbb{R}^{N \times L \times L} \quad (29)$$

where the transpose $k^{[1:L]^\top} \in \mathbb{R}^{N \times d \times L}$ interchanges the last two dimensions, the tensor multiplication $q^{[1:L]} k^{[1:L]^\top}$ is with respect to the last two dimensions, and the softmax is computed over the last dimension. Then

$$\tilde{\theta}^{[1:L]} = \alpha v^{[1:L]} \in \mathbb{R}^{N \times L \times d}, \quad (30)$$

where the tensor multiplication is over the last two dimensions.

C.2 TEMPORAL ATTENTION POSITION COMPONENT

Define the matrix of spatial graph representations for positions and its corresponding temporally-transformed matrix, across time:

$$\xi_i^{[1:L]} = \begin{bmatrix} \xi_i(1)^\top \\ \vdots \\ \xi_i(L)^\top \end{bmatrix} \in \mathbb{R}^{L \times n}, \tilde{\xi}_i^{[1:L]} = \begin{bmatrix} \tilde{\xi}_i(1)^\top \\ \vdots \\ \tilde{\xi}_i(L)^\top \end{bmatrix} \in \mathbb{R}^{L \times n}. \quad (31)$$

Similarly, construct the $h \times L \times L$ node-wise message-passing tensor:

$$M = \begin{bmatrix} \mathbf{m}_i(1, 1) & \dots & \mathbf{m}_i(1, L) \\ \vdots & \ddots & \vdots \\ \mathbf{m}_i(L, 1) & \dots & \mathbf{m}_i(L, L) \end{bmatrix}, \quad (32)$$

where $\mathbf{m}_i(t, s) \in \mathbb{R}^h$ is defined in Equation 9. Then we can write the update in Equation 9 as:

$$\tilde{\xi}_i^{[1:L]} = \xi_i^{[1:L]} + \xi_i^{[1:L]} \circ \mathbf{S} - \mathbf{T}, \quad (33)$$

where \odot is the Hadamard product and we define:

$$\mathbf{S} = \sum_{j=1}^L S_{ij} \in \mathbb{R}^L \quad (34)$$

$$[S]_{ij} = [\phi_{\text{inf}}(M) \circ (1_{L \times L} - I_{L \times L})]_{ij} \in \mathbb{R}$$

where $\phi_{\text{inf}}(M) \in \mathbb{R}^{L \times L}$ and

$$\begin{aligned} \mathbf{T} &= \sum_{j=1}^L T_{ij} \in \mathbb{R}^h \\ [T]_{ij} &= [\phi_{\text{inf}}(M) \circ (1_{L \times L} - I_{L \times L}) \circ \xi^{[1:L]}]_{ij} \in \mathbb{R}^h \\ \xi^{[1:L]} &= \begin{bmatrix} \xi_i(1) & \dots & \xi_i(L) \\ \vdots & \ddots & \vdots \\ \xi_i(1) & \dots & \xi_i(L) \end{bmatrix} \in \mathbb{R}^{L \times L \times n}. \end{aligned} \quad (35)$$

Again, we seek to find a differentiable expression that tensorizes the attention layer for all nodes. We stack $\xi_i^{[1:L]}$ for $i = 1, \dots, N$ into a tensor:

$$\tilde{\xi}^{[1:L]} := \begin{bmatrix} \tilde{\xi}_1^{[1:L]} \\ \vdots \\ \tilde{\xi}_N^{[1:L]} \end{bmatrix} = \begin{bmatrix} \tilde{\xi}_1^{[1:L]} \\ \vdots \\ \tilde{\xi}_N^{[1:L]} \end{bmatrix} + \begin{bmatrix} \tilde{\xi}_1^{[1:L]} \\ \vdots \\ \tilde{\xi}_N^{[1:L]} \end{bmatrix} \circ \mathbf{S} - 1_N \circ \mathbf{T}, \quad (36)$$

which is a differentiable function with respect to the $\xi_i(1), \dots, \xi_i(L)$ for $i = 1, \dots, N$, where $\tilde{\xi}^{[1:L]} \in \mathbb{R}^{N \times L \times n}$.

C.3 TEMPORAL ATTENTION VELOCITY COMPONENT

Define the matrix of spatial graph representations for velocities, across time:

$$\omega_i^{[1:L]} = \begin{bmatrix} \omega_i(1)^\top \\ \vdots \\ \omega_i(L)^\top \end{bmatrix} \in \mathbb{R}^{L \times n}. \quad (37)$$

We compute the weights as:

$$\beta_i = \text{softmax} \left(\frac{\omega_i^{[1:L]} \omega_i^{[1:L]\top}}{\sqrt{n}} \right) \in \mathbb{R}^{L \times L}. \quad (38)$$

Then we apply the weights to obtain the layer transform:

$$\tilde{\omega}_i^{[1:L]} = \beta_i \omega_i^{[1:L]} \in \mathbb{R}^{L \times n}. \quad (39)$$

We would like to find a tensorized expression that includes all nodes $i = 1, \dots, N$. Let $\omega^{[1:L]} = (\omega_1^{[1:L]}, \dots, \omega_N^{[1:L]}) \in \mathbb{R}^{N \times L \times n}$. As before, in a slight abuse of notation,

$$\beta = \text{softmax} \left(\frac{\omega^{[1:L]} \omega^{[1:L]\top}}{\sqrt{n}} \right) \in \mathbb{R}^{N \times L \times L}, \quad (40)$$

where the transpose $\omega^{[1:L]\top} \in \mathbb{R}^{N \times n \times L}$ interchanges the last two dimensions, the tensor multiplication $\omega^{[1:L]} \omega^{[1:L]\top}$ is with respect to the last two dimensions, and the softmax is computed over the last dimension. Hence,

$$\tilde{\omega}^{[1:L]} = \beta \omega^{[1:L]} \in \mathbb{R}^{N \times L \times n}, \quad (41)$$

where the tensor multiplication is over the last two dimensions.

C.4 TEMPORAL ATTENTION ADJACENCY COMPONENT

Let $A^{[1:L]} = (A(1), \dots, A(L)) \in \mathbb{R}^{L \times N \times N}$ be the tensor of all adjacency matrices over time. Let $k^{[1:L]} = A^{[1:L]}K$, $q^{[1:L]} = A^{[1:L]}Q$, $v^{[1:L]} = A^{[1:L]}V$ where $K, Q, V \in \mathbb{R}^{N \times N}$. With a slight abuse of notation, let

$$\pi = \text{softmax} \left(\frac{q^{[1:L]} k^{[1:L]\top}}{\sqrt{N}} \right) \in \mathbb{R}^{L \times N \times N}, \quad (42)$$

where the transpose $k^{[1:L]\top} \in \mathbb{R}^{L \times N \times N}$ interchanges the last two dimensions, the tensor multiplication $q^{[1:L]} k^{[1:L]\top}$ is with respect to the last two dimensions, and the softmax is computed over the last dimension as usual.

Thus,

$$\tilde{A}^{[1:L]} = \pi v^{[1:L]} \in \mathbb{R}^{L \times N \times N}. \quad (43)$$

D POSITIONAL ENCODING & LAYER NORMALIZATION

We use sinusoidal positional encodings, as defined in Vaswani et al. (2023). That is, we define the positional encodings $W^{[1:L]} \in \mathbb{R}^{L \times N \times d}$, $X^{[1:L]} \in \mathbb{R}^{L \times N \times n}$, $Y^{[1:L]} \in \mathbb{R}^{L \times N \times n}$, and $Z^{[1:L]} \in \mathbb{R}^{L \times N^2}$, used in Algorithm 1, as:

$$\begin{aligned} W_{i, \cdot, 2j} &= \sin \left(\frac{i}{\kappa^{2j/d}} \right), W_{i, \cdot, 2i+1} = \cos \left(\frac{i}{\kappa^{2j/d}} \right), \\ X_{i, \cdot, 2j} &= \sin \left(\frac{i}{\kappa^{2j/n}} \right), X_{i, \cdot, 2i+1} = \cos \left(\frac{i}{\kappa^{2j/n}} \right), \\ Y_{i, \cdot, 2j} &= \sin \left(\frac{i}{\kappa^{2j/n}} \right), Y_{i, \cdot, 2i+1} = \cos \left(\frac{i}{\kappa^{2j/n}} \right), \\ Z_{i, 2j} &= \sin \left(\frac{i}{\kappa^{2j/N^2}} \right), Z_{i, 2i+1} = \cos \left(\frac{i}{\kappa^{2j/N^2}} \right). \end{aligned} \quad (44)$$

We compress the adjacency matrix positional encoding as $Z^{[1:L]} \in \mathbb{R}^{L \times N^2}$ because we treat the N^2 entries of the adjacency matrix as belonging to a vectorspace, to which we apply a positional encoding.

Layer normalization in Algorithm 1 is defined as follows:

$$\begin{aligned} \hat{\mu}^{[1:L]} &= \begin{bmatrix} \frac{1}{d} \sum_{j=1}^d \theta_{1,j}^{[1:L]} \\ \vdots \\ \frac{1}{d} \sum_{j=1}^d \theta_{LN,j}^{[1:L]} \end{bmatrix} \in \mathbb{R}^{L \times N \times 1}, \\ \hat{\sigma}^{[1:L]} &= \begin{bmatrix} \sqrt{\frac{1}{d} \sum_{j=1}^d \left(\theta_{1,j}^{[1:L]} - \hat{\mu}_1^{[1:L]} \right)^2} \\ \vdots \\ \sqrt{\frac{1}{d} \sum_{j=1}^d \left(\theta_{LN,j}^{[1:L]} - \hat{\mu}_{LN}^{[1:L]} \right)^2} \end{bmatrix} \in \mathbb{R}^{L \times N \times 1}, \\ LN(\theta^{[1:L]}) &= \frac{\theta^{[1:L]} - \hat{\mu}^{[1:L]}}{\hat{\sigma}^{[1:L]}}, \end{aligned} \quad (45)$$

so we broadcast $\hat{\mu}, \hat{\sigma}$ across the second d dimensions of $\theta^{[1:L]}$. Likewise, for the adjacency matrix, we define:

$$\begin{aligned}\hat{\mu}^{[1:L]} &= \begin{bmatrix} \frac{1}{N^2} \sum_{j=1}^{N^2} \theta_{1,j}^{[1:L]} \\ \vdots \\ \frac{1}{N^2} \sum_{j=1}^{N^2} \theta_{L,j}^{[1:L]} \end{bmatrix} \in \mathbb{R}^{L \times 1}, \\ \hat{\sigma}^{[1:L]} &= \begin{bmatrix} \sqrt{\frac{1}{N^2} \sum_{j=1}^{N^2} \left(\theta_{1,j}^{[1:L]} - \hat{\mu}_1^{[1:L]} \right)^2} \\ \vdots \\ \sqrt{\frac{1}{N^2} \sum_{j=1}^{N^2} \left(\theta_{L,j}^{[1:L]} - \hat{\mu}_L^{[1:L]} \right)^2} \end{bmatrix} \in \mathbb{R}^{L \times 1}, \\ LN(A^{[1:L]}) &= \frac{A^{[1:L]} - \hat{\mu}^{[1:L]}}{\hat{\sigma}^{[1:L]}},\end{aligned}\tag{46}$$

where we view $A^{[1:L]} \in \mathbb{R}^{L \times N^2}$.

E CHARGED N -BODY DATASET: HYPER-PARAMETER SETTINGS & IMPLEMENTATION DETAILS

E.1 SET & SCALING RESULTS

After a hyper-parameter optimization for SET, we found the following optimal settings. Namely, we used an Adam optimizer with an initial learning rate of 4.45×10^{-5} , a batch size of 100, dropout of 0.1, 2 EGNN layers with SiLU activations, 3 vertical stacking layers with ReLU activations, hidden dimension of 128, spatial and temporal attention, and equivariance. The following hyper-parameters were inactive: weight decay, temporal adjacency, positional encoding and causal attention. We chose $\alpha = 1$ in Equation 14 and $B = 0.5$ in Equation 9.

The results in tables 3 and 4 show the effects of scaling the dataset. SET is the best-performing model across all datasets.

Model	Params	Test MSE
SET	796,058	5.72e-11
LSTM	1,269,833	1.02e-04
EGNN	100,612	5.34e-05
MLP	67,718	3.77e-06
Linear	3	4.48

Table 3: Performance metrics for optimized models with $N = 20$.

Model	Params	Test MSE
SET	796,058	1.20e-10
LSTM	1,859,513	1.95e-07
EGNN	100,612	5.88e-05
MLP	67,718	3.99e-06
Linear	3	5.19

Table 4: Performance metrics for optimized models with $N = 30$.

E.2 EGNN, LSTM, MLP, LINEAR BASELINES

For the EGNN baseline, we apply the same methodology outlined in Algorithm 1, whereby we apply MLPs $\phi_e, \phi_v, \phi_x, \phi_h$ across the time dimension:

$$\begin{aligned}
 \mathbf{h}_i^{(l+1)}(1) &= \phi_h(\mathbf{h}_i^{(l)}(1), \mathbf{m}_i(1)) \in \mathbb{R}^d \\
 &\vdots \\
 \mathbf{h}_i^{(l+1)}(L) &= \phi_h(\mathbf{h}_i^{(l)}(L), \mathbf{m}_i(L)) \in \mathbb{R}^d,
 \end{aligned} \tag{47}$$

for $l = 1, \dots, K$. We used an Adam optimizer without weight decay and with an initial learning rate of 3.98×10^{-5} , batch size of 32, hidden dimension of 64, 3 EGNN layers, and residual connections.

For the LSTM baseline, we found using only temporal attention was optimal. That is, we did not pre-process the graph data with an EGNN; we simply applied a vanilla LSTM to the graph data. Each token embedding $\mathbf{z}(t) \in \mathbb{R}^{N \times (d+2n+2N-2)}$ comprises of the following data:

$$\begin{aligned}
 \mathbf{h}_1(t), \dots, \mathbf{h}_N(t) &\in \mathbb{R}^{N \times d} \\
 \mathbf{x}_1(t), \dots, \mathbf{x}_N(t) &\in \mathbb{R}^{N \times n} \\
 \mathbf{v}_1(t), \dots, \mathbf{v}_N(t) &\in \mathbb{R}^{N \times n} \\
 \mathbf{e}_{0,1}(t), \dots, \mathbf{e}_{0,N}(t), \mathbf{e}_{1,0}(t), \dots, \mathbf{e}_{1,N}(t), \\
 \dots, \mathbf{e}_{N,0}(t), \dots, \mathbf{e}_{N,N-1}(t) &\in \mathbb{R}^2.
 \end{aligned} \tag{48}$$

We used Adam with an initial learning rate of 2.47×10^{-5} , batch size of 32, hidden dimension of 512, 2 EGNN layers, 3 temporal stacking layers, and dropout of 0.1. We omitted the use of weight decay, temporal adjacency, and recurrence. The MLP uses 5 hidden layers with hidden dimension 128. It only takes a tensor of positions $x \in \mathbb{R}^{L \times N \times n}$ and velocities $v \in \mathbb{R}^{L \times N \times n}$ as inputs. It is trained using an Adam optimizer with a learning rate of 1.75×10^{-5} and a batch size of 100. The linear dynamics model simply predicts position $x(t) = x(t-1) + \alpha v(t-1)$ and velocity $v(t) = \beta v(t-1) + \gamma$, learning α, β, γ using an Adam optimizer with initial learning rate of 2.73×10^{-5} , weight decay of 1×10^{-6} , and batch size of 100.

F CLASSICAL N -BODY DATASET: HYPER-PARAMETER SETTINGS & IMPLEMENTATION DETAILS

F.1 DATASET: GRAVITATIONAL MASSES

Masses exert gravitational forces on each other according to Newton’s universal law of gravitation. The force \mathbf{F}_i acting on mass m_i in an N -body system is given by Trenti & Hut (2008):

$$\mathbf{F}_i = - \sum_{j \neq i} \frac{G m_i m_j}{|\mathbf{x}_i - \mathbf{x}_j|^2} \frac{\mathbf{x}_i - \mathbf{x}_j}{|\mathbf{x}_i - \mathbf{x}_j|} - \nabla \cdot \phi(\mathbf{x}_i) \tag{49}$$

where G is the universal gravitational constant and ϕ is an external scalar potential. Solving the forward dynamics of the system involves a system of non-linear second order ordinary differential equations:

$$\frac{\partial^2 \mathbf{x}_i}{\partial t^2} = \frac{\mathbf{F}_i}{m_i}. \quad (50)$$

Hence, by the Picard-Lindelöf theorem, the initial value problem with specified initial positions $\mathbf{x}_i(0)$ and initial velocities $\mathbf{v}_i(0) = \frac{\partial \mathbf{x}_i}{\partial t}(0)$ has a unique solution. However, this system only has an analytic solution up to $N = 2$ Trenti & Hut (2008). Thus, larger systems require advanced numerical integration techniques. Figure 3 shows how a system with $N = 20$ bodies evolves over 10 seconds, obeying the law of energy conservation.

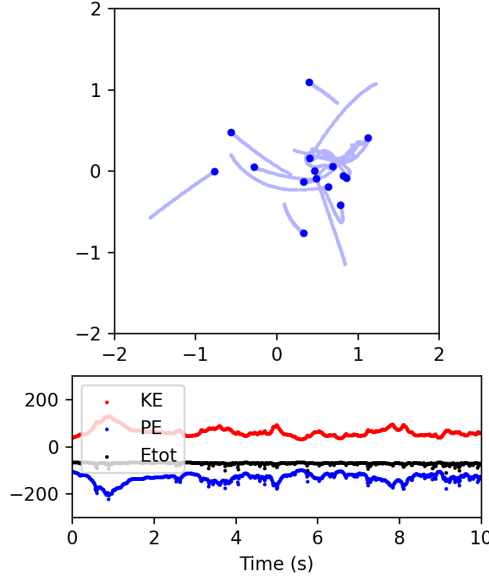


Figure 3: Evolution of a 20-body system over 10 seconds, with plots of kinetic energy, potential energy, and total energy in Joules.

We use the N -body system simulator from Mocz (2020) to generate a dataset with $100k$ trajectories without an external scalar potential ϕ . We sample $80k$ trajectories for training, $10k$ trajectories for validation, and $10k$ trajectories for testing. The model is fed graph data for $L = 10$ time steps, sampled 10 apart, and predicts the trajectory at a horizon of $H = 1k$ timesteps. We consider $N = 20$ masses, whereby positions, velocities, and masses are known at each timestep. As before, the edge attributes between gravitational masses are $e_{ij}(t) = (m_i m_j, \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\|_2^2)$ and the features are $h_i(t) = \|\mathbf{v}_i(t)\|_2$ for $i = 1, \dots, 5$. We select the best model according to a hyper-parameter optimization with 30 trials, as per Appendix F.

F.2 ABLATION STUDY: EQUIVARIANCE, ADJACENCY, AND ATTENTION

We conduct an ablation study on the use of equivariance, temporal adjacency, and spatial and temporal attention in SET, shown in Table 5. By selecting the best model on the validation set, we find that incorporating equivariance, adjacency, and only spatial attention improves performance. Unlike in the charged N -body problem, temporal attention is not useful for this dataset.

Ablation	Model	Params	Val MSE	Test MSE	MSE Ratio
Attention	Equiv=True, Adj=True, SATT=True, TATT=False	175,882	1.84e1	1.84e1	—
	Equiv=True, Adj=True, SATT=True, TATT=True	185,488	2.50e1	2.52e1	1.37×
Adjacency	Equiv=True, Adj=False , SATT=True, TATT=True	175,888	3.19e1	3.22e1	1.75×
Equivariance	Equiv=False , Adj=True, SATT=True, TATT=True	185,620	3.71e1	3.77e1	2.05×

Table 5: Ablation study of attention, adjacency, and equivariance for $N = 20$.

F.3 BASELINES & SCALING N

We performed model selection using a hyper-parameter optimization on SET. The best settings found are as follows: an Adam optimizer with an initial learning rate of 5×10^{-5} , a batch size of 100, dropout of 0.1, 3 EGNN layers with SiLU activations, 4 vertical stacking layers with ReLU activations, hidden dimension of 32, only spatial attention, and equivariance imposed. We report results with both spatial and temporal attention to highlight the effects of scaling on the bonafide SET architecture. The following hyper-parameters were inactive: weight decay, temporal adjacency, positional encoding and causal attention. We chose $\alpha = 1$ in Equation 14.

Model	Params	Test MSE
SET	175,888	7.36
EGNN	34,409	4.61
EGNN SchNet	73,098	4.72
$SE(3)$ -Transformer	395,972	11.93

Table 6: Performance metrics for models with $N = 20$.

We compare our best performing SET model with EGNN, EGNN SchNet, and $SE(3)$ -Transformer baselines for a varying number of masses $N = 5, 20, 50$. EGNN outperforms all baselines, as seen in Table 7. Indeed, it appears that the imposition of temporal structure is not useful for this dataset, unlike the charged N -body dataset. SET performs the best when N is large; nonetheless, the number of parameters grows like $O(N^2)$ because we use temporal adjacency.

Model	N	Params	Val MSE	Test MSE
EGNN	5	34,409	2.04	2.05
	20	34,409	4.57	4.61
	50	34,409	4.65	4.65
EGNN SchNet	5	73,098	2.13	2.12
	20	73,098	4.66	4.72
	50	73,098	4.73	4.74
$SE(3)$ -Transformer	5	395,972	77.71	78.60
	20	395,972	11.93	12.04
	50	395,972	8.60	8.71
SET	5	175,888	31.89	32.20
	20	3,641,488	7.36	7.37
	50	144,235,888	7.00	7.05

Table 7: Scaling the number of masses $N = 5, 20, 50$.

F.4 SCALING L

For SET, we fix $N = 5$ and scale the sequence length for $L = 10, 50, 100$, as shown in table 8. Evidently, shorter sequences L yield better test MSE.

Sequences length L	Test MSE
5	25.20
50	115.83
100	322.29

Table 8: Scaling $L = 10, 50, 100$ in SET, with $N = 5$ masses.

Thus, we find that SET performs most optimally when N is large and L is small.