

A predictive machine learning force-field framework for liquid electrolyte development

In the format provided by the
authors and unedited

Table of Contents

1 Supplementary Note: Details of Initial Training by DFT Data	2
1.1 Graph Equivariant Transformer	2
1.2 Separation of Interactions	4
1.3 Charge Equilibrium	6
1.4 Initial Training of BAMBOO	11
1.5 DFT Dataset Construction	13
2 Supplementary Note: Ensemble Knowledge Distillation for MLFF	18
2.1 Theory	18
2.2 Case Study of Liquid Electrolytes	19
2.3 Case Study of Solid Phase Transformation	21
3 Supplementary Note: Density Alignment	23
3.1 Theory	23
3.2 Performance	24
4 Supplementary Note: Ablation Study of Model Architecture	30
5 Supplementary Note: Details of Molecular Dynamics	34
6 Supplementary Note: Comparison with Other Force Fields	35
7 Supplementary Note: Properties of Simple Liquid Electrolytes	35
8 Supplementary Note: Results of More Practical Liquid Electrolytes	40
9 Supplementary Note: Transferability to Unseen Molecules	43
10 Supplementary Note: More Solvation Structure Analyses	45
11 Supplementary Note: Liquid Electrolyte Data	48
References	48

1 Supplementary Note: Details of Initial Training by DFT Data

1.1 Graph Equivariant Transformer

Given atomic structures, we first initialize node scalar representation x_i^0 and node vector representation \vec{V}_i^0 for atom i based on the atom type z_i , and initialize edge scalar representation d_{ij} and edge vector representation \vec{e}_{ij} based on the relative vector between atom i and atom j (\vec{r}_{ij}) within a certain cut off radius r_{cut} (5 Å in this work):

$$x_i^0 = \text{embeddingMLP}(z_i), \quad (\text{S1})$$

$$\vec{V}_i^0 = \vec{0}, \quad (\text{S2})$$

$$\hat{r}_{ij} = \frac{\vec{r}_{ij}}{\|\vec{r}_{ij}\|}, \quad (\text{S3})$$

$$d_{ij} = \text{SiLU}(W^d \text{RBF}(\cos \frac{\pi \|r_{ij}\|}{r_{\text{cut}}} + 1)), \quad (\text{S4})$$

$$\vec{e}_{ij} = d_{ij} * \hat{r}_{ij}. \quad (\text{S5})$$

Here, embeddingMLP is a multi-layer perceptron (MLP) that maps discrete atom types into continuous embeddings implemented in the NN.EMBEDDING class in PyTorch [1], $\|\cdot\|$ is the L2 norm of a vector, RBF is the radial basis functions-based expansion as in Ref. [2] that expands distance into high-dimensional representations, and SiLU refers to the Sigmoid-Weighted Linear Units [3] activation function. In equation (S4), W^d refers to the weight matrix used to generate d_{ij} . In this work, we define “ $*$ ” as the following multiplication operation to combine a scalar $A \in R^m$ and a vector $\vec{B} \in R^{n \times l}$, $n = 1$ or m , which keeps equivariance of \vec{B} .

$$A * B = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_m \end{bmatrix} * \begin{bmatrix} B_{11}, B_{12}, \dots, B_{1l} \\ B_{21}, B_{22}, \dots, B_{2l} \\ \vdots \\ B_{n1}, B_{n2}, \dots, B_{nl} \end{bmatrix} = \begin{bmatrix} A_1 B_{11}, A_1 B_{12}, \dots, A_1 B_{1l} \\ A_2 B_{21}, A_2 B_{22}, \dots, A_2 B_{2l} \\ \vdots \\ A_m B_{n1}, A_m B_{n2}, \dots, A_m B_{nl} \end{bmatrix}. \quad (\text{S6})$$

After initialization, we input x_i^0 , \vec{V}_i^0 , d_{ij} , \vec{e}_{ij} into the GET layers. In the n^{th} layer, we first input x_i^n into a LayerNorm [4] layer, then we input x_i^n , $\|r_{ij}\|$, and d_{ij} into the classic QKV-transformer shown in Figure 1e in the main text and described as below:

$$q_i^n = W^{q,n} x_i^n + b^{q,n}, \quad (\text{S7})$$

$$k_i^n = W^{k,n} x_i^n + b^{k,n}, \quad (\text{S8})$$

$$v_i^n = W^{v,n} x_i^n + b^{v,n}. \quad (\text{S9})$$

Here q_i^n , k_i^n , and v_i^n are queries, keys and values in transformer [5] (notice that capital V stands for vector embedding whereas lowercase v stands for “values” in QKV-attention). We further compute the attention weight a_{ij}^n and the intermediate node scalar representation y_i^n . Furthermore, we build an intermediate vector representation \vec{u}_i^n to incorporate edge vector representation \vec{e}_{ij} into the GET layer and sum contributions of neighboring atoms to the vector representation.

$$a_{ij}^n = \text{SiLU}(\langle q_i^n, k_j^n \rangle) (\cos \frac{\pi ||r_{ij}||}{r_{\text{cut}}} + 1), \quad (\text{S10})$$

$$y_i^n = \sum_j a_{ji}^n (v_j^n \odot d_{ij}), \quad (\text{S11})$$

$$\vec{u}_i^n = \sum_j v_j^n * \vec{e}_{ij}. \quad (\text{S12})$$

Here, \odot is element-wise multiplication. For the vector embeddings, we first update node vector representation \vec{V}_i^n by linear projection:

$$\vec{U}_i^{1,n} = W^{U^1,n} \vec{V}_i^n, \quad (\text{S13})$$

$$\vec{U}_i^{2,n} = W^{U^2,n} \vec{V}_i^n, \quad (\text{S14})$$

$$\vec{U}_i^{3,n} = W^{U^3,n} \vec{V}_i^n. \quad (\text{S15})$$

Then, we use inner product $\langle \cdot, \cdot \rangle$ to generate the scalar representation $w_i^n \in R^m$ from $\vec{U}_i^{1,n}, \vec{U}_i^{2,n} \in R^{m \times 3}$, which is consequently used to interact with the intermediate node scalar representation y_i^n :

$$w_i^n = \langle \vec{U}_i^{1,n}, \vec{U}_i^{2,n} \rangle. \quad (\text{S16})$$

Before final output, we further transform y_i^n into three parts:

$$O_i^{1,n} = W^{O^1,n} y_i^n + b^{O^1,n}, \quad (\text{S17})$$

$$O_i^{2,n} = W^{O^2,n} y_i^n + b^{O^2,n}, \quad (\text{S18})$$

$$O_i^{3,n} = W^{O^3,n} y_i^n + b^{O^3,n}. \quad (\text{S19})$$

Finally, we output the updated node scalar representation x_i^{n+1} and node vector representation \vec{V}_i^{n+1} for the $(n+1)^{th}$ layer:

$$x_i^{n+1} = x_i^n + w_i^n \odot O_i^{2,n} + O_i^{3,n}, \quad (\text{S20})$$

$$\vec{V}_i^{n+1} = \vec{V}_i^n + O_i^{1,n} * \vec{U}_i^{3,n} + \vec{u}_i^n. \quad (\text{S21})$$

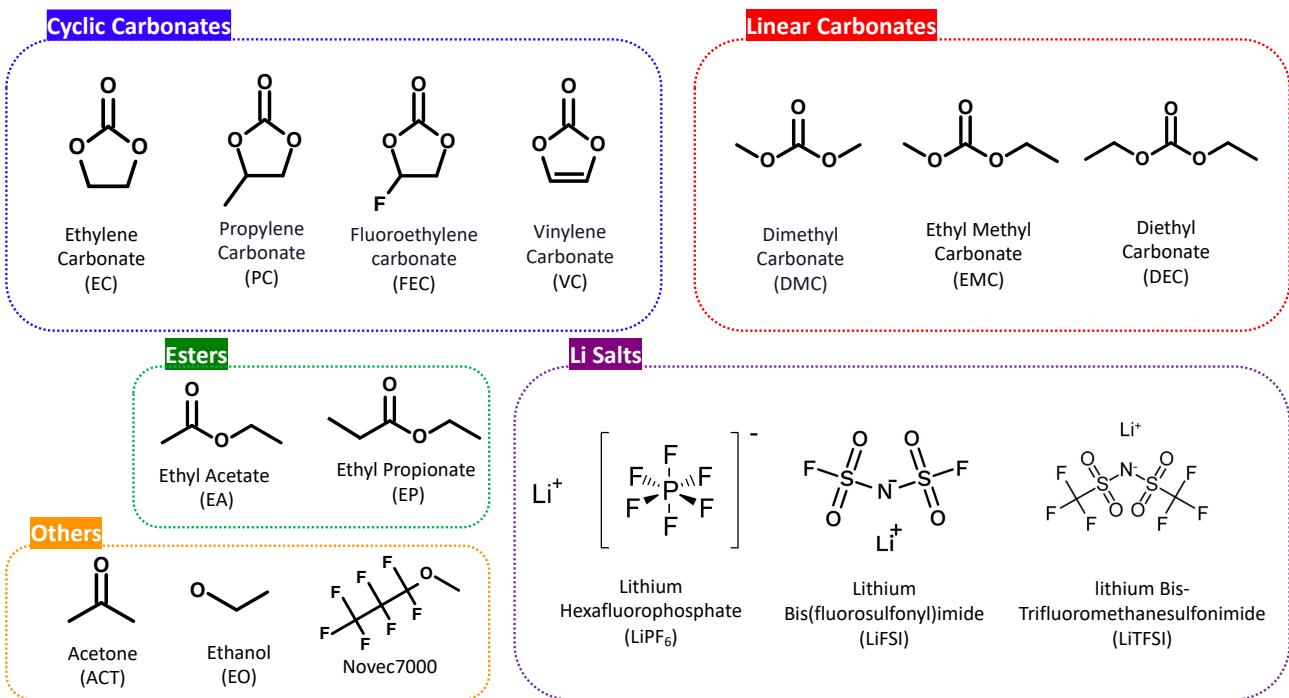


Figure Supplementary 1: Schematic of the species included in the DFT dataset.

As above, both of the node scalar and vector representations in the $(n+1)^{th}$ layer encode both scalar and vector information from both node and edge in the n^{th} layer.

In this work, we refrain from calculating many-body features like bond angles and dihedral angles for two main reasons. Firstly, the GET layer in our model inherently captures all many-body effects [6] as defined by the derivative-based understanding of many-body interactions. In other words, many-body interactions are implicitly represented in BAMBOO. Secondly, referencing the findings from another study [7], introducing explicit three-body terms into transformer-based graph neural networks does not significantly enhance the prediction of properties of atomic structures, yet it substantially increases computational demand. Additionally, we avoid using tensor products to maintain fast inference speed, given that tensor products are known for their slow speed.

1.2 Separation of Interactions

Once we reach the final GET layer (the n^{th} layer below, and $n=3$ in this work), we compute NN-based per atom energy E_i^{NN} and atomic partial charge q_i based on x_i^n by two MLPs:

$$E_i^{\text{NN}} = \text{MLP}^{\text{energy}}(x_i^n), \quad (\text{S22})$$

$$q_i = \text{MLP}^{\text{charge}}(x_i^n), \quad (\text{S23})$$

and we compute NN-based forces by:

$$\vec{f}_{ij}^{\text{NN}} = -\frac{\partial E^{\text{NN}}}{\partial r_{ij}} + \frac{\partial E^{\text{NN}}}{\partial r_{ji}}, \vec{f}_i^{\text{NN}} = \sum_{j \neq i} \vec{f}_{ji}^{\text{NN}}. \quad (\text{S24})$$

Then, we define the electrostatic energy model based on the charge equilibrium theory [8]:

$$E^{\text{elec}} = \sum_i \chi_i q_i + \frac{1}{2} \sum_{i,j} \frac{q_i q_j}{r_{ij}} + \frac{1}{2} \sum_i J_i q_i^2. \quad (\text{S25})$$

Here, χ_i is the element-wise electronegativity, and J_i is the element-wise electronic hardness. The reason behind why we do not predict χ and J based on structures is stated in the proof of Theorem 1.1 in the next section. In this work, both of the two quantities only depend on atomic types and are learned and predicted by MLPs:

$$\chi_i = (\text{MLP}^\chi(x_i^0))^2, \quad (\text{S26})$$

$$J_i = (\text{MLP}^J(x_i^0))^2. \quad (\text{S27})$$

Here we use the square of the outputs of the two MLPs to ensure that both of the two quantities are positive. More discussion about charge equilibrium is provided in the next section.

In addition, given atomic structures, we can directly compute dispersion interaction (E_i^{disp} and \vec{f}_i^{disp}) by the D3 correction [9]. Note that we do not include the dispersion correction in the DFT training data and therefore do not consider E_i^{disp} and \vec{f}_i^{disp} in the whole training process. We only incorporate dispersion correction in BAMBOO during MD simulations. Finally, we combine NN, electrostatic, and dispersion interactions to compute the total energy and force of each atom:

$$E_i^{\text{total}} = E_i^{\text{NN}} + E_i^{\text{elec}} + E_i^{\text{disp}}, \quad (\text{S28})$$

$$\vec{f}_i^{\text{total}} = \vec{f}_i^{\text{NN}} + \vec{f}_i^{\text{elec}} + \vec{f}_i^{\text{disp}}. \quad (\text{S29})$$

We further compute the virial tensor \mathbf{T} based on pairwise forces:

$$\vec{f}_{ij} = \vec{f}_{ij}^{\text{NN}} + \vec{f}_{ij}^{\text{elec}} + \vec{f}_{ij}^{\text{disp}}, \quad (\text{S30})$$

$$\mathbf{T} = \sum_i^N \sum_j^N \vec{f}_{ij} \otimes \vec{r}_{ij}, \quad (\text{S31})$$

where \otimes is the outer product between the two 3-dimensional vectors. Here, according to equation (S24) and (S33) and the nature of $\vec{f}_{ij}^{\text{disp}}$, $\vec{f}_{ij} = -\vec{f}_{ji}$, which satisfies the Newton's Third Law of opposite forces for computing the microscopic stress [10], and since we compute \vec{r}_{ij} across periodic boundary, we do not need additional correction for computing \mathbf{T} of periodic systems. We also compute the dipole moment as $\vec{D} = \sum_{i=1}^N q_i \vec{r}_i$ for training the machine learning model.

In this work, we use a cosine cutoff function as in equation (S4) and equation (S10) to ensure that the expanded distance feature d_{ij} and the attention score a_{ij}^n smoothly decay to zero at the cutoff distance r_{cut} . Since all mathematical operations used in this work are smooth and differentiable, the outputs of the model

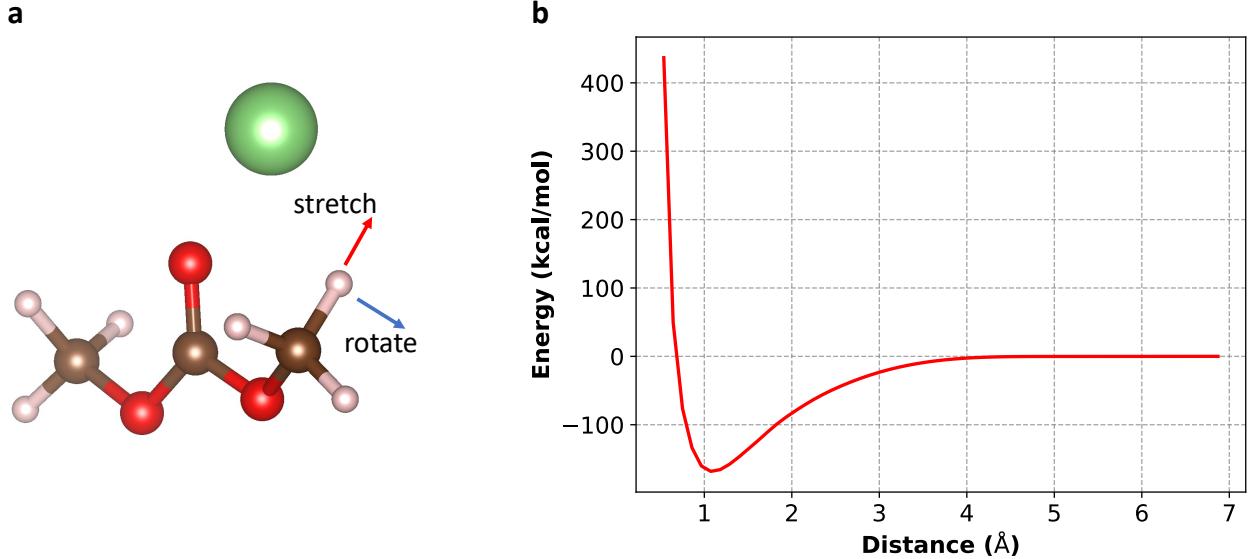


Figure Supplementary 2: **Smoothness of the potential energy surface.** **a** Illustration of a cluster with a Li ion and a DMC molecule. Color coding: Li: green; C: brown; O: red; H: pink. **b.** Predicted energy versus distance between the stretched H atom and the closest C atom. The stretching direction is along the bond vector between C and H as in **a**.

should also be smooth and differentiable. To demonstrate smoothness of the potential energy surface, we plot the energy change of C-H bond stretching in a cluster with a Li ion and a DMC molecule as in Figure [Supplementary 2a](#). As we can see in [Figure Supplementary 2b](#), the energy of the cluster changes smoothly from less than 1 Å to 7 Å which demonstrates the smoothness of potential energy surface from a unphysically small distance to a unphysically large distance.

1.3 Charge Equilibrium

It is well-known that a fixed-charge model may impose serious restrictions on the fidelity and transferability of force fields. However, allowing charges to vary throughout a molecular dynamics simulation is non-trivial. Partial charge is an abstraction of electron density and should reflect the fact that electron density is at electrostatic equilibrium at any moment. If we arbitrarily change the charge in a MD simulation, we are actually pumping energy into (or dissipating energy from) the system and violating the basic energy conservation law that is necessary to collect correct statistical results from the simulations. Therefore, the charges should be allowed to change to reflect the necessary physics but also restricted to obey the physical laws. This is the central topic of polarizable force fields, and many different approaches have been developed in the past decades for different systems [11, 12].

In this work, we incorporate the charge equilibrium model (Qeq hereafter) into our MLFF for its efficiency. We further use a regularization strategy to make our GNN model directly output the Qeq charges such that an expensive iterative equilibrium solver per MD timestep is completely removed. By definition, the force (exact energy derivative) associated with the electrostatic energy should be:

$$\vec{f}_i^{\text{elec}*} = -\frac{\partial E^{\text{elec}}}{\partial r_i} = -\sum_j \frac{q_i q_j}{r_{ij}^2} \hat{r}_{ij} + \sum_i \chi_i \frac{\partial q_i}{\partial r_i} + \frac{1}{2} \sum_{i,j} \frac{1}{r_{ij}} \frac{\partial(q_i q_j)}{\partial r_i} + \frac{1}{2} \sum_i J_i \frac{\partial q_i^2}{\partial r_i}. \quad (\text{S32})$$

Although for non-periodic systems, $\partial E^{\text{elec}}/\partial r_i$ is straightforward to compute by auto-differentiation in PyTorch [1], for periodic boxes in MD simulations, most MD engines such as LAMMPS [13], can only compute Ewald summation [14] for the long-range electrostatic force defined as below:

$$\vec{f}_i^{\text{elec}} = -\sum_j \frac{q_i q_j}{r_{ij}^2} \hat{r}_{ij}. \quad (\text{S33})$$

Technically, it is difficult to explicitly compute the remaining terms in equation (S32) as it is expensive to compute $\partial q_j/\partial r_i$ for all the pairs of atoms. To bypass such difficulty, we refer to the charge equilibrium theory [8].

In the following, we demonstrate that, under charge equilibrium, $\vec{f}_i^{\text{elec}} = \vec{f}_i^{\text{elec}*}$. To simplify the notations in the proof, in the following we introduce vector $\mathbf{x} = \{\chi_1, \dots, \chi_N\}, \mathbf{x} \in R^N$, and matrix $\mathbf{J} : J_{ij} = 1/r_{ij}, J_{ii} = J_i, J \in R^{N \times N}$. We prove the following two theorems.

Theorem 1.1 *If $\mathbf{q} = \{q_1, \dots, q_N\}, q \in R^N$, such that*

$$\mathbf{q} = \arg \min_{\mathbf{q}} (E(\mathbf{q}) - \lambda(\sum_i q_i - q_{\text{total}})), \quad (\text{S34})$$

where λ is the Lagrange multiplier to ensure that $\sum_i q_i = q_{\text{total}}$, then

$$\vec{f}_i^{\text{elec}} = -\frac{\partial E^{\text{elec}}}{\partial r_i} = -\sum_j \frac{q_i q_j}{r_{ij}^2} \hat{r}_{ij}. \quad (\text{S35})$$

Proof of Theorem A.1 We first define the target of minimization:

$$L(\mathbf{q}, \lambda) = \mathbf{x}^T \mathbf{q} + \frac{1}{2} \mathbf{q}^T \mathbf{J} \mathbf{q} - \lambda(\mathbf{1}^T \mathbf{q} - q_{\text{total}}), \quad (\text{S36})$$

where $\mathbf{1}$ is an all-one vector. If \mathbf{q} is a minimizer of L , then

$$0 = \frac{\partial L}{\partial \mathbf{q}} = \mathbf{x} + \frac{1}{2} \mathbf{J}^T \mathbf{q} + \frac{1}{2} \mathbf{J} \mathbf{q} - \lambda \mathbf{1} = \mathbf{x} + \mathbf{J} \mathbf{q} - \lambda \mathbf{1}, \quad (\text{S37})$$

where the second “=” is because \mathbf{J} is symmetric. Similarly, we also have:

$$0 = \frac{\partial L}{\partial \lambda} = q_{\text{total}} - \mathbf{1}^T \mathbf{q}. \quad (\text{S38})$$

Now we calculate $-\partial E^{\text{elec}}/\partial r_i$. To simplify the writing by avoiding 3-dimensional tensors, here we consider just

one dimension of \vec{r}_i , for example, z_i ,

$$\frac{\partial E}{\partial z_i} = \frac{\partial}{\partial z_i} (\mathbf{x}^T \mathbf{q} + \frac{1}{2} \mathbf{q}^T \mathbf{J} \mathbf{q}) = \frac{\partial \mathbf{x}^T}{\partial z_i} \mathbf{q} + \mathbf{x}^T \frac{\partial \mathbf{q}}{\partial z_i} + \frac{1}{2} \frac{\partial \mathbf{q}^T}{\partial z_i} \mathbf{J} \mathbf{q} + \frac{1}{2} \mathbf{q}^T \frac{\partial \mathbf{J}}{\partial z_i} \mathbf{q} + \frac{1}{2} \mathbf{q}^T \mathbf{J} \frac{\partial \mathbf{q}}{\partial z_i}. \quad (\text{S39})$$

Here, the term $\frac{\partial \mathbf{x}^T}{\partial z_i} \mathbf{q} = 0$, because χ does not depend on structures. The terms $\frac{1}{2} \frac{\partial \mathbf{q}^T}{\partial z_i} \mathbf{J} \mathbf{q}$ and $\frac{1}{2} \mathbf{q}^T \mathbf{J} \frac{\partial \mathbf{q}}{\partial z_i}$ can be combined into $\mathbf{q}^T \mathbf{J} \frac{\partial \mathbf{q}}{\partial z_i}$ due to the symmetry of \mathbf{J} . The fourth term $\frac{1}{2} \mathbf{q}^T \frac{\partial \mathbf{J}}{\partial z_i} \mathbf{q}$ is the desired expression of \vec{f}_i^{elec} .

For the remaining terms, we have:

$$\mathbf{x}^T \frac{\partial \mathbf{q}}{\partial z_i} + \mathbf{q}^T \mathbf{J} \frac{\partial \mathbf{q}}{\partial z_i} = (\mathbf{x} + \mathbf{J} \mathbf{q})^T \frac{\partial \mathbf{q}}{\partial z_i} = \lambda \mathbf{1}^T \frac{\partial \mathbf{q}}{\partial z_i} = \lambda \frac{\partial (\mathbf{1}^T \mathbf{q})}{\partial z_i} = \lambda \frac{\partial q_{\text{total}}}{\partial r_i} = 0. \quad (\text{S40})$$

where the second “=” is from equation (S37), the fourth “=” is from equation (S38), and the last “=” is because q_{total} is a constant for a given system. Therefore, we have proved that:

$$\vec{f}_i^{\text{elec}} = -\frac{\partial E}{\partial z_i} = -\frac{1}{2} \mathbf{q}^T \frac{\partial \mathbf{J}}{\partial z_i} \mathbf{q}. \quad (\text{S41})$$

Then, we prove the reverse theorem:

Theorem 1.2 *If $\mathbf{q} = \{q_1, \dots, q_N\}, q \in R^N$ such that it satisfies equation (S41), and $\text{rank}(\frac{\partial \mathbf{q}}{\partial c}) = N - 1, c_i \in \{x_1, y_1, z_1, \dots, x_N, y_N, z_N\}$, then there exists λ such that $\mathbf{q} = \arg \min_{\mathbf{q}} (E(\mathbf{q}) - \lambda(\sum_i q_i - q_{\text{total}}))$.*

Proof of Theorem A.2 Combining equation (S39), (S41) and the fact that $\frac{\partial \mathbf{x}^T}{\partial z_i} \mathbf{q} = 0$, we have:

$$\mathbf{x}^T \frac{\partial \mathbf{q}}{\partial z_i} + \mathbf{q}^T \mathbf{J} \frac{\partial \mathbf{q}}{\partial z_i} = (\mathbf{x} + \mathbf{J} \mathbf{q})^T \frac{\partial \mathbf{q}}{\partial z_i} = 0, \quad (\text{S42})$$

which can extend to all $c_i \in \{x_1, y_1, z_1, \dots, x_N, y_N, z_N\}$. We then combine all $\frac{\partial \mathbf{q}}{\partial c_i}$ into a matrix with the dimensions of $3N \times N$, denoted as $\frac{\partial \mathbf{q}}{\partial \mathbf{c}}$. Because $\sum_i q_i = q_{\text{total}}$, therefore for all c_i , we have $\mathbf{1}^T \frac{\partial \mathbf{q}}{\partial c_i} = 0$, and consequently $\frac{\partial \mathbf{q}^T}{\partial \mathbf{c}} \mathbf{1} = \mathbf{0}$. As a result, the rank of $\frac{\partial \mathbf{q}}{\partial \mathbf{c}}$ is at most $N - 1$. If $\text{rank}(\frac{\partial \mathbf{q}}{\partial \mathbf{c}}) = N - 1$ as in the theorem, then due to equation (S42), $\mathbf{x} + \mathbf{J} \mathbf{q}$ is in the linear space spanned by $\mathbf{1}$. Therefore, there exists λ such that $\mathbf{x} + \mathbf{J} \mathbf{q} = \lambda \mathbf{1}$, which also means $\frac{\partial L(\mathbf{q})}{\partial \mathbf{q}} = 0$ where L is defined in equation (S36). Because $\frac{\partial^2 L(\mathbf{q})}{\partial \mathbf{q}^2} = \mathbf{J}$, where J is a diagonally-dominant matrix with all positive diagonal elements, $\frac{\partial^2 L(\mathbf{q})}{\partial \mathbf{q}^2}$ is a positive-definite matrix, therefore $L(\mathbf{q})$ is a positive definite quadratic form of \mathbf{q} , and \mathbf{q} is the global minimizer of $L(\mathbf{q})$.

From theorem 1.1, we know that the predicted charge \mathbf{q} should satisfy equation (S34) to ensure that \vec{f}_i^{elec} is strictly conservative. However, in practice, to satisfy equation (S34), the time complexity to compute matrix \mathbf{J} is $\mathcal{O}(N^2)$, and the time complexity to compute the inverse of \mathbf{J} is $\mathcal{O}(N^3)$. Since the time complexity of the GET layers is $\mathcal{O}(N)$, computing \mathbf{J} and \mathbf{J}^{-1} would largely slow down the MD simulation for large systems with thousands of atoms. For example, in the very recent work by Hu *et al.* [15], it is shown that the cost of calculating the full Qeq for a system with 5,000 atoms is around 1 second per step per GPU, which equals to 90 ps per day per 5,000 atoms (with the time step of 1 fs as this work). We can see that the speed of calculating full Qeq is much slower than the BAMBOO model (5 ns per day per 5,000 atoms) and not practical

for simulating liquid electrolytes where thousands of atoms and nanoseconds of MD simulation are required. Moreover, since there is energy dissipation in the NPT and NVT ensembles used for studying liquid electrolytes in MD, small energy drift by non-conservative force would not affect results from MD simulations. In Ref. [16], the authors use a recursive method to approximate equation (S34) with the time complexity of $\mathcal{O}(N)$. However, this recursive method would also slow down the inference speed, the degree of which depends on the number of iterations: the closer approximation, the more iterations, and the slower inference speed.

In this work, we design a method to approximate charge equilibrium in equation (S34) with the minimum increase of inference speed, in which we decouple the inference speed with the degree of approximation. Instead of additional operations at inference, we move the approximation to the training process by adding a loss term in equation (S46) to minimize the difference between \vec{f}_i^{elec} and $\vec{f}_i^{\text{elec}*}$, which is an approximation to charge equilibrium as in Theorem 1.2:

$$L_{\text{Qeq}} = \|\vec{f}_i^{\text{elec}} - \vec{f}_i^{\text{elec}*}\|^2. \quad (\text{S43})$$

Although this method seems straightforward as it directly minimizes the difference between the electrostatic force we can compute and the derivative of the electrostatic energy with respect to coordinates, equation (S43) actually has the physical meaning of charge equilibrium as in the Theorems 1.1 and 1.2. The degree of approximation depends on the magnitude of the weight factor of α_{Qeq} in the total loss as below. We show the importance of α_{Qeq} in Figure [Supplementary 3a](#), where smaller weight factors results in significant energy drift, while the largest weight factor only leads to 0.02 kcal/mol/atom energy increase in the 800 ps NVE simulation. This degree of energy drift is negligible in MD simulations with the NVT and NPT ensemble, as there is energy dissipation in this two ensembles. In Figure [Supplementary 3b](#) and [c](#), we show that the temperature of 1 ns NVT MD simulation and 4 ns NPT simulation slightly fluctuates around the set temperature of the simulation (303 K), and pressure around one atmosphere, which demonstrates the stability of the MD simulation. This method of charge equilibrium regularization minimally increases the inference speed by adding two summation over atoms as in equation (S25).

To more clearly demonstrate the role of L_{Qeq} on approximating energy derivative, we further compare the difference between \vec{f}_i^{elec} and $\vec{f}_i^{\text{elec}*}$ and the different microscopic behavior and bulk observables from MLFFs with different α_{Qeq} . At the beginning of this project, we did not consider L_{Qeq} ($\alpha_{\text{Qeq}} = 0$), because we thought that if we train the MLFF well, then the predicted force should be close to DFT force and everything would be fine. However, we found that MD simulations containing DMC and Li salts might randomly crash because of the spurious C-H bond breaking event at the methyl group of DMC. This phenomenon is not trivial because the energy landscape of the C-H bond from the MLFF is reasonable as in Figure [Supplementary 2b](#). After analyzing the MD trajectory, we realized that the C-H bond breaking was a result of randomly happened fast spin of the methyl group that is close to a Li ion, as illustrated in Figure [Supplementary 2a](#).

With the background, we show the effect of including $\alpha_{\text{Qeq}} L_{\text{Qeq}}$ in the loss term on the cluster in Figure [Supplementary 2a](#). We first directly compare \vec{f}_i^{elec} and $\vec{f}_i^{\text{elec}*}$. From Table [Supplementary 1](#), we can see that larger

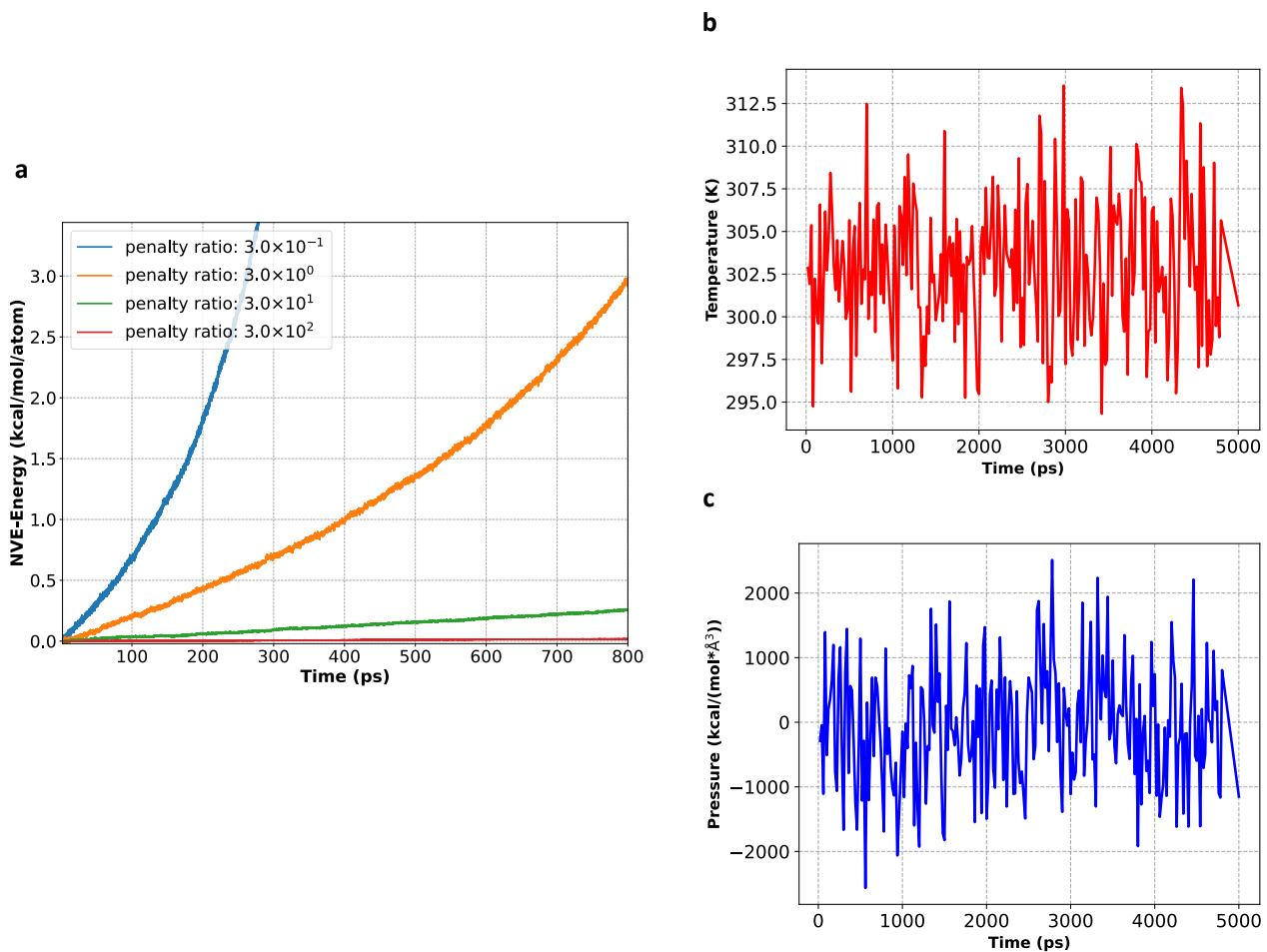


Figure Supplementary 3: **a** Energy drift in molecular dynamics simulations by NVE ensemble. The simulated system is EC|LiPF₆|0.95. **b** and **c** Temperature and pressure versus simulation time of DMC_EC|51.49|LiFSI|0.49, respectively.

α_{Qeq} , smaller $\|\vec{f}_i^{\text{elec}} - \vec{f}_i^{\text{elec}*}\|$, which clearly demonstrates the effectiveness of approximating the exact energy derivative. Then, we consider two quantities related to the rotation of the methyl H atom, accumulated work W_{circle} and momentum M_{circle} from one circle of rotation:

$$W_{\text{circle}} = \int_{\text{circle}} \vec{f}_i^{\text{elec}} \cdot d\vec{r}, \quad (\text{S44})$$

$$M_{\text{circle}} = \int_{\text{circle}} \vec{f}_i^{\text{elec}} \times \vec{r}_{C-H}. \quad (\text{S45})$$

In principle, W_{circle} and M_{circle} should be zero if the force is the exact derivative of the electrostatic energy. If not, then the accumulated energy and momentum might drive the H atom to spin. As in Table [Supplementary 1](#), larger α_{Qeq} , smaller W_{circle} and M_{circle} , and consequently, when α_{Qeq} reaches 30 and 300, we do not observe spin of the methyl group and C-H bond breaking during the 5 ns MD simulations. Out of an abundance of caution, in this work we set α_{Qeq} to be 300, and during all the production MD simulations in this work, there is no spin of methyl group and spurious bond breaking event.

As for the bulk properties from MLFFs trained with different α_{Qeq} , from Table [Supplementary 1](#) we do not observe a clear trend between α_{Qeq} and density ρ and Li diffusivity D_{Li} , although we can see that smaller α_{Qeq} , larger fluctuation of bulk properties. Since we calculate ρ and D_{Li} by trajectories before C-H bond breaking, intuitively the shortening of trajectories introduces additional noise into the computation of ρ and D_{Li} . The independence of the mean of bulk properties on α_{Qeq} might be because, the predicted force does not largely depend on α_{Qeq} as force itself is a label included in the training and finetuning process. In other words, the performance of prediction of force and downstream observables is mainly determined by whether the MLFF learns DFT labels and experimental labels well. As in Table [Supplementary 1](#), even when $\alpha_{Qeq} = 0$, $\|\vec{f}_i^{\text{elec}} - \vec{f}_i^{\text{elec}*}\|$ is just 0.178 kcal/(mol*Å), which is 8.1% of the force prediction RMSE (2.19 kcal/(mol*Å)), and 4.7% of the magnitude of DFT force of the H atom (3.76 kcal/(mol*Å)). We can also see that, $\|\vec{f}_i^{\text{elec}} - \vec{f}_i^{\text{elec}*}\|$ is smaller than $\|\vec{f}_i^{\text{BAMBOO}} - \vec{f}_i^{\text{DFT}}\|$, except when $\alpha_{Qeq} = 0$ where the two quantities are comparable. Therefore, we argue that the degree of approximating exact energy derivative does not largely affect the downstream observables, because the difference between predicted force and exact energy derivative is smaller than the difference between predicted force and target force label.

1.4 Initial Training of BAMBOO

Given the design of the GET architecture and the prediction of energy, force, virial and dipole, we use the following loss function to train the neural networks in BAMBOO:

Table Supplementary 1: Comparison between behaviors of MLFFs trained with different α_{Qeq} . Forces, work, and momentum are calculated for the methyl H atom marked in Figure [Supplementary 2a](#). Spin of methyl group, C-H bond breaking, density, and Li diffusivity are measured from the 5 ns MD simulations of the DMC_EC|70_30|LiPF6|1.0 system, and density ρ and Li diffusivity D_{Li} are calculated by trajectories before C-H bond breaking. Units: f : (kcal/(mol*Å)); W and M : (kcal/mol); ρ : (g/cm³); D_{Li} : (10^{-10} m²/s).

Behaviors	$\alpha_{Qeq} = 0$	0.3	3	30	300
$\ \vec{f}_i^{\text{elec}} - \vec{f}_i^{\text{elec}*}\ $	0.178	0.107	0.054	0.020	0.007
$\ \vec{f}_i^{\text{BAMBOO}} - \vec{f}_i^{\text{DFT}}\ $	0.180	0.290	0.240	0.302	0.139
W_{circle}	$8.9 * 10^{-3}$	$2.8 * 10^{-3}$	$2.1 * 10^{-3}$	$1.0 * 10^{-3}$	$5.5 * 10^{-4}$
M_{circle}	0.105	0.036	0.022	0.007	0.001
Spin of methyl group	Yes	Yes	Yes	No	No
C-H bond breaking	Yes	Yes	Yes	No	No
ρ	1.219 ± 0.021	1.228 ± 0.015	1.217 ± 0.009	1.216 ± 0.004	1.221 ± 0.003
D_{Li}	2.401 ± 0.246	2.378 ± 0.155	2.523 ± 0.132	2.542 ± 0.073	2.435 ± 0.061

$$\begin{aligned}
L_{\text{training}} = & \frac{1}{N_{\text{cluster}}} \sum_k^{N_{\text{cluster}}} \sum_i^{N_{\text{atom}}} (\alpha_{\text{energy}}^{\text{training}} (E_k^{\text{DFT}} - E_k^{\text{BAMBOO}})^2 \\
& + \frac{1}{N_{\text{atom}}} \alpha_{\text{force}}^{\text{training}} \|\vec{f}_{k,i}^{\text{DFT}} - \vec{f}_{k,i}^{\text{BAMBOO}}\|^2 \\
& + \alpha_{\text{virial}}^{\text{training}} \|\mathbf{T}_k^{\text{DFT}} - \mathbf{T}_k^{\text{BAMBOO}}\|^2 \\
& + \frac{1}{N_{\text{atom}}} \alpha_{\text{charge}}^{\text{training}} (q_{k,i}^{\text{DFT}} - q_{k,i}^{\text{BAMBOO}})^2 \\
& + \alpha_{\text{dipole}}^{\text{training}} \|\vec{D}_k^{\text{DFT}} - \vec{D}_k^{\text{BAMBOO}}\|^2 \\
& + \frac{1}{N_{\text{atom}}} \alpha_{\text{Qeq}}^{\text{training}} \|\vec{f}_{k,i}^{\text{elec, BAMBOO}} - \vec{f}_{k,i}^{\text{elec*, BAMBOO}}\|^2),
\end{aligned} \tag{S46}$$

where α_{energy} is the weight associated with energy in the loss function, and so on. Here, q^{DFT} is the CHELPG [17], which is a partial charge that is fitted to the electrostatic potential outside the cluster. From equation (S46), we can see that the charge learned by BAMBOO is determined by four aspects: the first is to fit energy, force, and virial together with the contribution from the neural network; the second is to fit the electrostatic potential by fitting CHELPG; the third is to fit dipole moment, a macroscopic observable derived from charge densities by DFT; and the fourth is to approximate charge equilibrium as in equation (S43). In Figure [Supplementary 4a, b, and c](#), we show that BAMBOO can learn and predict DFT computed energy, force, and virial with R^2 scores higher than 0.99.

As in Figure [Supplementary 5a](#) and [b](#), we can see that both dipole moment and electrostatic potential, two observables largely affecting inter-molecular interactions, can be predicted with R^2 scores higher than 0.98. For dipole moment, we observe some outliers in Figure [Supplementary 5a](#), which are mostly clusters with delocalized electrons, such as those with Li clusters. In Figure [Supplementary 5c](#) we show the relation between BAMBOO-predicted charge and CHELPG [17], the charge derived by fitting the electrostatic potential from DFT. The RMSE between BAMBOO and CHELPG is around 0.15 a.u., and the R^2 score is around 0.85, which is a decent score but lower than that of dipole moment and electrostatic potential. More importantly, we can see

that there are some horizontal lines in Figure [Supplementary 5c](#), indicating series of structures where CHELPG varies considerably while BAMBOO provides more consistent charge predictions. Here, we argue that the higher degree of variability of CHELPG might not be physical due to the high numerical instability of the charge fitting process [18]. For example, in Figure [Supplementary 5d](#) we show the partial charge of a H atom from different sources versus the degree of rotation of the methyl group in the DMC molecule as in Figure [Supplementary 2a](#). We can see that CHELPG changes raggedly with respect to the rotation, which does not appear physical, while BAMBOO's predictions remain stable during the rotation, which aligns with Figure [Supplementary 5c](#) that BAMBOO provides more consistent charge predictions. In Figure [Supplementary 5d](#) we also show the Mulliken charge [19] and the MBIS charge [20], two popular DFT-based partial charges. For the Mulliken charge, although it also changes considerably during the rotation, such variation is much smoother than CHELPG. For the MBIS charge, it remains quite stable during the rotation, which is similar to BAMBOO. Therefore, the smaller degree of variability of charge predicted by BAMBOO compared with CHELPG cannot be criticized as a failure of BAMBOO. We will study the effect of using different DFT-based partial charges to train MLFFs on simulating different chemical systems. The high stability of BAMBOO-predicted charges also implies that the changes of partial charge reported in Figure 3 in the main text are not due to numerical instability of prediction, but come from the inherent difference of atomic environments captured by BAMBOO. Fundamentally speaking, since the concept of atomic partial charge cannot well capture delocalized charge density distribution, in future works we will try to tackle this issue through more sophisticated representations of charge density, and the concept of Wannier functions[21] might be an option.

Here, we build the neural networks in BAMBOO by PyTorch [1], and we use the AdamX optimizer [22] with the learning rate of 10^{-2} . In the training process, weight decay is set to 0.001, while the learning rate decreases at a rate of 0.99. The batch size is 128 for both training and validating. The training process is composed of 750 training epochs. The model consists of 3 layers, with 32 radial basis functions for edges and an embedding dimension of 64 for nodes. It features 16 attention heads, employs a cutoff radius of 5.0 Å for each message passing layer, and includes the MLPs with two layers for predicting energy and charge. The loss weights α 's in equation (S46) are provided as the default in the open source codes at [23] and listed in Table [Supplementary 2](#).

Table Supplementary 2: Table of loss weights.

Loss weights	$\alpha_{\text{energy}}^{\text{training}}$	$\alpha_{\text{force}}^{\text{training}}$	$\alpha_{\text{virial}}^{\text{training}}$	$\alpha_{\text{charge}}^{\text{training}}$	$\alpha_{\text{dipole}}^{\text{training}}$	α_{Qeq}
Value	0.01	0.3	0.01	10.0	10.0	300.0

1.5 DFT Dataset Construction

In this work, we use 720 thousand clusters with the DFT calculated energy, atomic forces, virial tensor, dipole moment, and partial charge (CHELPG) to train BAMBOO. The calculations are performed utilizing the B3LYP [24] functional alongside the def2-svpd [25] basis set, supplemented by density fitting techniques

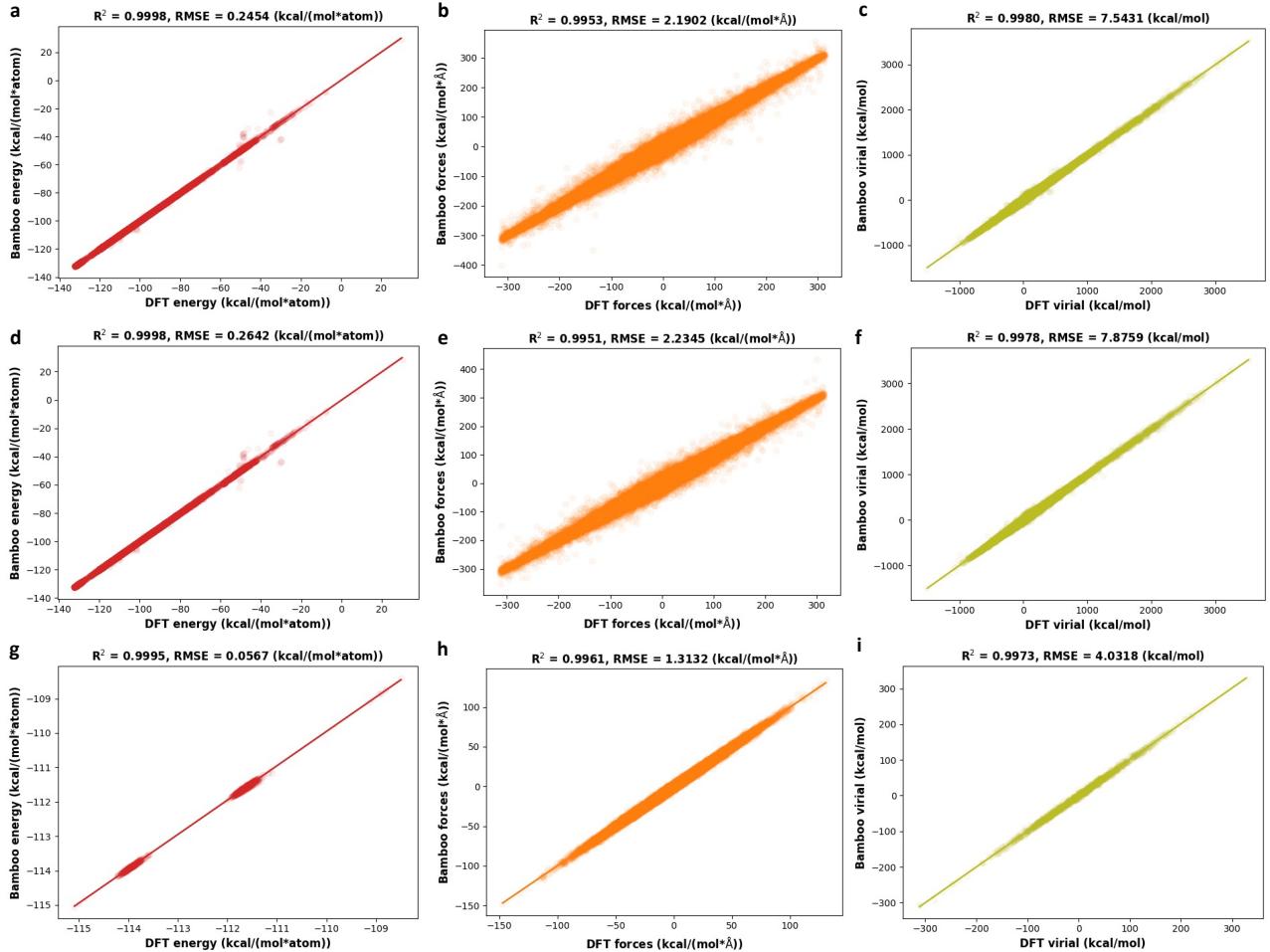


Figure Supplementary 4: Fitting performance of BAMBOO before and after ensemble knowledge distillation and density calibration. **a, b, c.** DFT computed energy, force, and virial versus BAMBOO predicted quantities before ensemble knowledge distillation and density calibration, respectively. **d, e, f.** DFT computed energy, force, and virial versus BAMBOO predicted quantities after ensemble knowledge distillation and density calibration, respectively. The results are based on the validation set. **g, h, i.** DFT computed energy, force, and virial versus BAMBOO predicted quantities for the test set sampled from the production MD simulations, respectively.

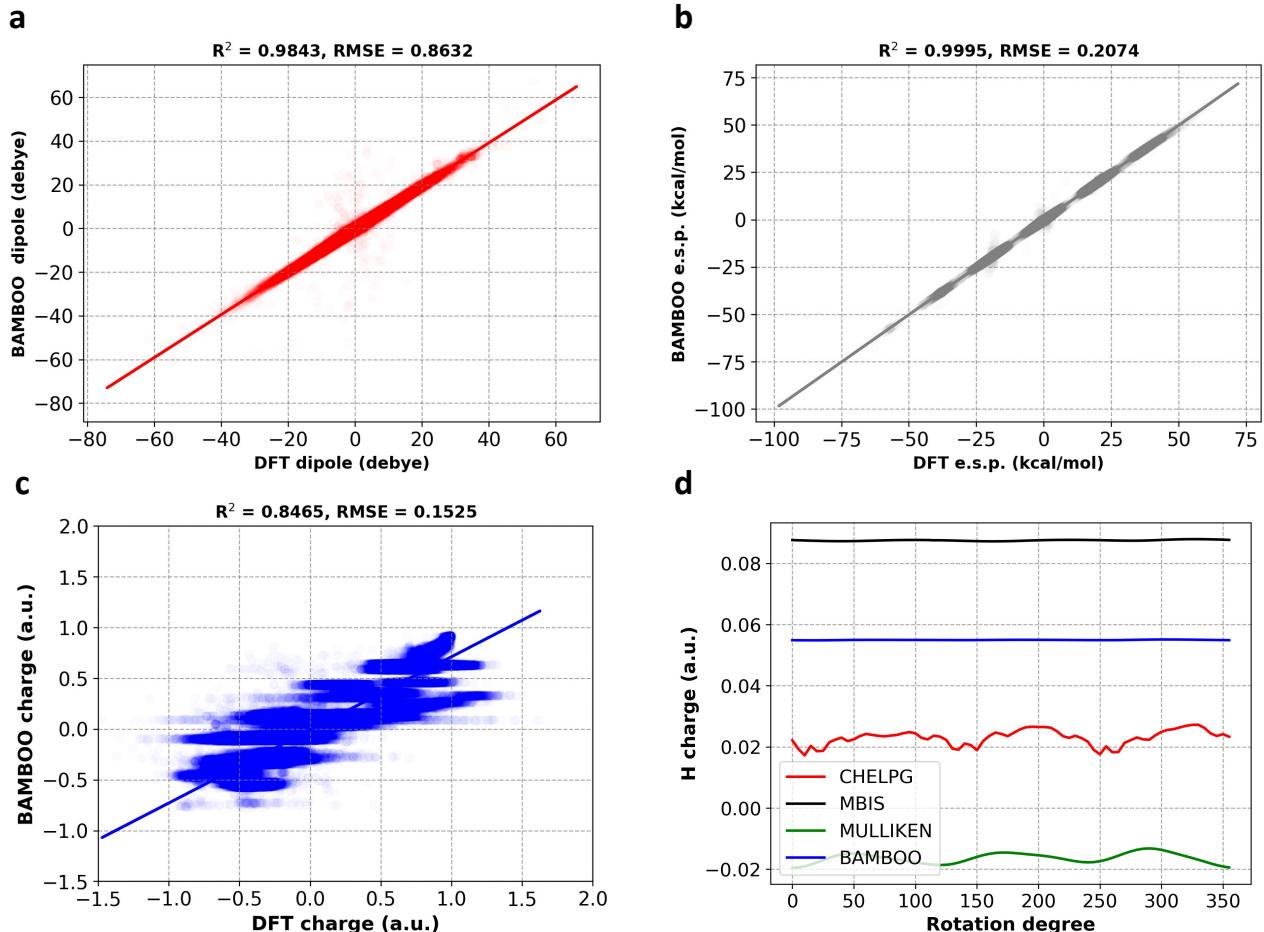


Figure Supplementary 5: **Dipole moment, electrostatic potential, and charge by DFT and BAMBOO**
a, b. DFT computed dipole moment and electrostatic potential (abbreviated as "e.s.p.") versus BAMBOO predicted quantities, respectively. Electrostatic potential is sampled at grid points 10 Å away from the geometric center of the cluster. **c** BAMBOO predicted charge versus CHELPG for the validation set. The results are based on the DFT validation set. **d** Rotation degree of the methyl group versus charge of H predicted by BAMBOO and three DFT-based charge partition methods.

based on the def2-universal-jkfit auxiliary basis [26]. Throughout the computational process, the convergence threshold for the self-consistent field is set to 1.0×10^{-10} a.u., with a maximum allowance of 100 iterations. Here, we use the open-source GPU4PySCF package [27, 28] developed by ByteDance Inc. for the DFT calculations. Compared with QChem [29] on 32 CPUs, GPU4PySCF on a single V100 GPU can save up to 97% computation time and 95% cost.

The procedure of collecting cluster structures is described below: first we take clusters in Ref. [30] as the starting point, which contains species including EC, PC, VC, FEC, DMC, EMC, DEC, PF_6^- , and Li^+ . Then, we train five BAMBOO models by the DFT labels of the initial cluster structures calculated by GPU4PySCF, and run MD simulations using one randomly picked model at 300K for 1 ns with the NPT ensemble on pure solvents of EA, EP, EO, ACT, and Novec7000, and solutions with $\text{EC:DMC:EMC:PC:LiPF}_6 = 3:3:1:1:1$, $\text{EC:DMC:EMC:VC:LiFSI} = 3:3:1:1:1$, and $\text{EC:DMC:LiFSI:LiTFSI} = 10:10:1:1$ by moles. In the first round of MD simulations, we observe that spurious reactions (mainly bond breaking) would occur in most simulations with new molecules and anions, which is expected as the initial structures do not cover the chemistry of the new species, especially FSI^- and TFSI^- . Then, we sample clusters from the MD trajectories until the first spurious reaction happens, and we use the following active learning scheme to sample clusters from the MD trajectories.

1. Instead of focusing on the uncertainty of the energy of the cluster as discussed in Ref. [30], our attention shifts to the uncertainty of predictions concerning atomic properties \vec{f}_i and q_i . First, we employ five models to compute the standard deviation of predictions of charge σ_q and magnitude of force $\sigma_{||\vec{f}||}$ on each atom. We then select all atoms for which $\sigma_q > 0.015$ a.u. and $\sigma_{||\vec{f}||} > 3$ kcal/(mol·Å). For hydrogen atoms, we lower the criteria to $\sigma_q > 0.01$ a.u. and $\sigma_{||\vec{f}||} > 2$ kcal/(mol·Å) as hydrogen is significantly lighter than other elements and typically exhibits smaller magnitudes of q and \vec{f} .
2. When encountering uncertain atoms (expect Li^+), we first determine the molecule/anion to which the uncertain atom belongs, and consider the single molecule/anion as a cluster.
3. Next, we search for molecules, anions, and cations that have at least one atom within a 5 Å radius of the uncertain atom. This search includes the molecule/anion to which the uncertain atom belongs. We consider all the neighboring molecules/anions/cations found in this search as a cluster. It's worth noting that if multiple atoms within a single molecule are identified as uncertain, we perform the search for neighboring molecules only once to minimize redundancy.
4. We input clusters containing no more than 110 atoms into GPU4PySCF for labelling, and incorporate the newly labeled clusters and DFT labels into the dataset for training.
5. We repeat steps 1-4 iteratively. Once a system demonstrates stability without spurious reactions during the 1 ns MD simulation, we terminate the procedure for that particular system.

We achieve stable MD simulations for all the pure solvents within two iterations, and for solutions within three iterations. In Figure [Supplementary 6](#), we present the distribution of number of atoms in clusters and the number of molecules in the datasets. We observe two peaks in cluster sizes: one with fewer atoms corresponding

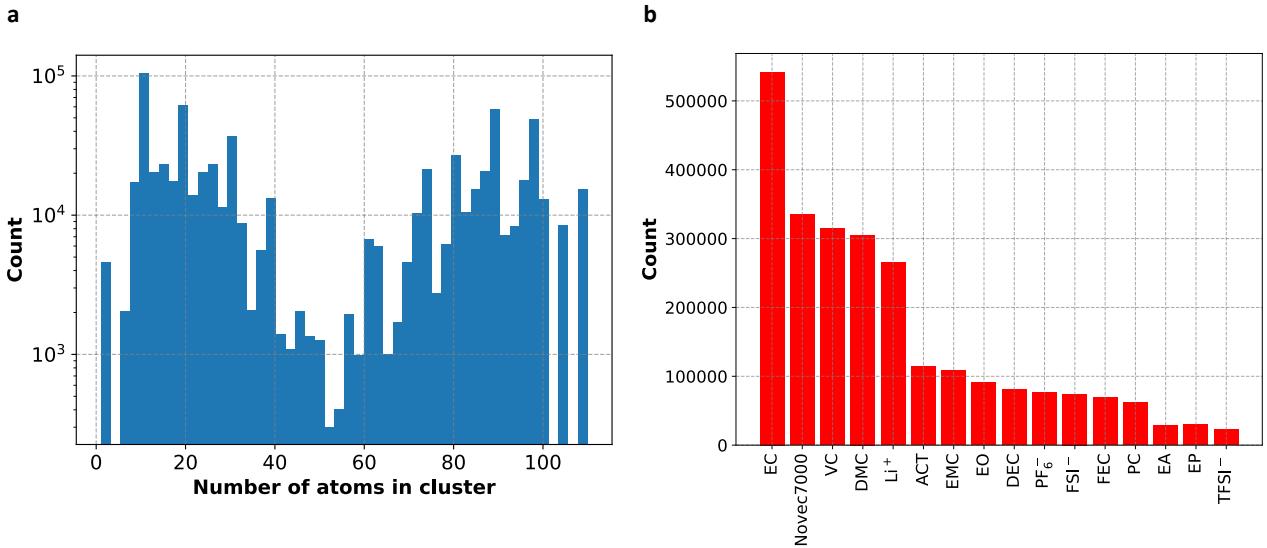


Figure Supplementary 6: **a** and **b** Count of number of atoms in a cluster and number of molecules/ions in the dataset, respectively.

to single molecules/anions, and another with more atoms representing the complete clusters within 5 Å of the central uncertain atoms. Regarding the distribution of molecules, EC, Novec7000, VC, and DMC are the four most frequently sampled molecules. Among these, EC and DMC are the two main solvents of interest in this study, while VC and Novec7000 possess unique structural features compared with other molecules (such as a carbon-carbon double bond in VC and a high ratio of fluorination in Novec7000).

After each iteration of data collection, we randomly choose 90% data as the training set, and 10% as the validation set to evaluate the performance of the training process. Therefore, the validation set has the same distribution as the training set where there are considerable number of high energy structures such as distorted molecules, decomposed molecules, and even clusters with only Li. Although these off-equilibrium structures are necessary to stabilize the MD simulations driven by the MLFFs, such high energy states might boost the prediction RMSE. Since Figure [Supplementary 4a](#) to [f](#) are all based on the validation set, the RMSE values reported at those figures can be further lowered if we only consider near-equilibrium structures that would appear in production MD simulations. Similar to Dajnowicz *et al.* [30] where the test set is separately sampled from production MD simulations, we construct a separate test set sampling 10,000 clusters from production MD simulations, and we use the MLFF used for production (the one after density alignment) to predict the energy, force, and virial of the test set. As in Figure [Supplementary 4g](#), [h](#) and [i](#), we can see that the RMSE of energy, force, and virial predictions on the test set decrease 78%, 41%, and 49% compared with the validation set, which more closely reflect the degree of prediction errors in the production MD simulations.

2 Supplementary Note: Ensemble Knowledge Distillation for MLFF

2.1 Theory

As noted in Ref. [31], MLFFs trained by different random seeds often lead to fluctuating results from MD simulations, particularly when extrapolating to systems not seen in the training set. In Figure [Supplementary 7a](#), we illustrate that even if the ensemble of MLFFs is well-trained and behaves consistently and accurately within the interpolation domain, the inherent randomness of neural networks might lead to greater variability in out-of-domain inference compared with in-domain inference [32]. Consequently, the ensemble of MLFFs may exhibit significantly different behaviors within the extrapolation domain. In Figure [Supplementary 7b](#), we propose a method to constrain the behavior of different MLFFs within the extrapolation domain by training them on the mean predictions. This approach not only mitigates differences among the ensemble of MLFFs within the extrapolation domain but also reduces MLFF errors by minimizing random errors.

We illustrate the reduction of random errors by expressing the predicted energy in terms of the true energy (or other quantities such as force and virial tensor), systematic error, and random error.

$$E_{\text{pred}}^{\text{initial}} = E_{\text{true}} + \delta E_{\text{systematic}}^{\text{initial}} + \delta E_{\text{random}}^{\text{initial}}, \quad (\text{S47})$$

where δE denotes the error of energy. We can write the mean of the predicted energy as:

$$\overline{E_{\text{pred}}^{\text{initial}}} = E_{\text{true}} + \overline{\delta E_{\text{systematic}}^{\text{initial}}} + \overline{\delta E_{\text{random}}^{\text{initial}}}. \quad (\text{S48})$$

According to the law of large number, if the mean is taken over sufficient amount of predictions, then we have $\overline{\delta E_{\text{systematic}}^{\text{initial}}} \approx \delta E_{\text{systematic}}^{\text{initial}}$ and $\overline{\delta E_{\text{random}}^{\text{initial}}} \approx 0$. Therefore, we have:

$$\overline{E_{\text{pred}}^{\text{initial}}} \approx E_{\text{true}} + \delta E_{\text{systematic}}^{\text{initial}}, \quad (\text{S49})$$

which implies

$$|\overline{E_{\text{pred}}^{\text{initial}}} - E_{\text{true}}| < |E_{\text{pred}}^{\text{initial}} - E_{\text{true}}|. \quad (\text{S50})$$

In ensemble knowledge distillation, since we use $\overline{E_{\text{pred}}^{\text{initial}}}$ as the label to train the MLFF, we can write the predicted energy after ensemble knowledge distillation as:

$$\begin{aligned} E_{\text{pred}}^{\text{distilled}} &= \overline{E_{\text{pred}}^{\text{initial}}} + \delta E_{\text{systematic}}^{\text{distill}} + \delta E_{\text{random}}^{\text{distill}} \\ &\approx E_{\text{true}} + \delta E_{\text{systematic}}^{\text{initial}} + \delta E_{\text{systematic}}^{\text{distill}} + \delta E_{\text{random}}^{\text{distill}}. \end{aligned} \quad (\text{S51})$$

Therefore, if we properly design the training process in ensemble knowledge distillation to minimize $\delta E_{\text{systematic}}^{\text{distill}}$ and $\delta E_{\text{random}}^{\text{distill}}$, we will achieve a smaller difference between E_{true} and $E_{\text{pred}}^{\text{distilled}}$ than $E_{\text{pred}}^{\text{initial}}$. In the case study of solid phase transformation, we will demonstrate that following ensemble knowledge distillation, the prediction

error decreases.

2.2 Case Study of Liquid Electrolytes

In this study, as we train MLFFs using DFT data of small gas-phase clusters and utilize them to predict properties of bulk liquid electrolytes, we essentially task MLFFs with extrapolation, which may introduce significant random noise into MD simulations. Hence, we adopt the following procedure to conduct ensemble knowledge distillation for liquid electrolytes:

1. Initially, we train five models purely by DFT data at random.
2. Subsequently, we randomly select one model to run 600 ps NPT MD simulations and 400 ps NVT MD simulations for all systems shown in Figure [Supplementary 7c](#).
3. With the trajectories obtained from the MD simulations from the previous step, we employ all five models to predict the energy, forces, and virial tensor of 100 randomly selected frames from the trajectory of each system.
4. Finally, we use the mean predictions as the labels to finetune the randomly selected model by the following loss function:

$$L_{\text{distill}} = L_{\text{training}} + \frac{1}{N_{\text{frame}}} \sum_k^{N_{\text{frame}}} \sum_i^{N_{\text{atom}}} \alpha_{\text{energy}}^{\text{distill}} (\overline{E_k^{\text{BAMBOO}}} - E_k^{\text{BAMBOO}})^2 + \frac{1}{N_{\text{atom}}} \alpha_{\text{force}}^{\text{distill}} \|\overline{f_{ki}^{\text{BAMBOO}}} - \bar{f}_{ki}^{\text{BAMBOO}}\|^2 + \alpha_{\text{virial}}^{\text{distill}} \|\overline{\mathbf{T}_k^{\text{BAMBOO}}} - \mathbf{T}_k^{\text{BAMBOO}}\|^2. \quad (\text{S52})$$

Here, we incorporate L_{training} into L_{distill} to regulate the behavior of the distilled MLFF on DFT data. Specifically, in this study, we only fine-tune the weights of $\text{MLP}^{\text{energy}}$ of the MLFF, and we solely include the energy, forces, and virial terms in L_{training} within equation (S52). This approach is adopted because we do not include L_{Qeq} in L_{distill} , as it is challenging to compute $\partial E^{\text{elec}} / \partial r_i$ for periodic bulk systems. Consequently, we refrain from adjusting weights that would impact the prediction of charge to maintain the constraint of charge equilibrium introduced during the initial training. Note that, to compare the effect of ensemble distillation on reducing standard deviations of MD results from different models in Figure [Supplementary 7c](#), we conduct distillation on all five models to calculate standard deviations from the five models. However, in the density alignment section below, we utilize only the randomly selected model to minimize computational costs.

As a result of ensemble knowledge distillation, we observe that the errors of predictions on DFT data remain almost unchanged (when L_{training} is not included in L_{distill} , the increase in error can be up to 30%). Additionally, we find that the deviation between predictions of the randomly selected model and ensemble mean predictions is nearly halved. For systems with 4,000 atoms, the root mean square error (RMSE) between $\overline{E_k^{\text{BAMBOO}}}$ and E_k^{BAMBOO} decreases from approximately 0.41 kcal/mol/atom to 0.23 kcal/mol/atom, RMSE of forces decreases

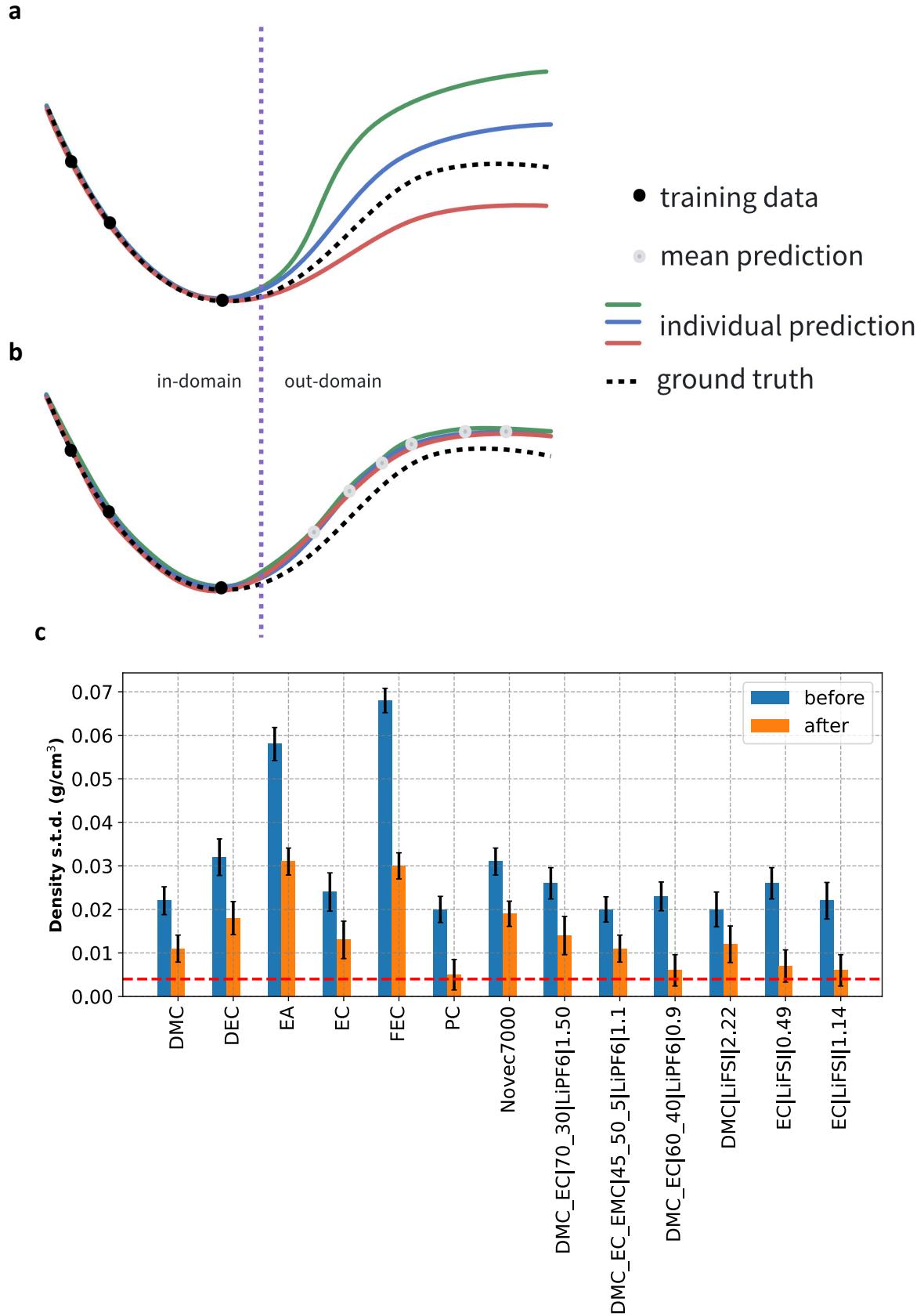


Figure Supplementary 7: **a** and **b** Illustration of the interpolation and extrapolation behaviors before and after ensemble knowledge distillation, respectively. **c** Standard deviations of density from MD by five MLFFs before and after ensemble knowledge distillation. Error bars are defined as the mean density fluctuations during each MD run, which accounts for how much density standard deviation comes from the intrinsic fluctuation of MD by the same force field model.

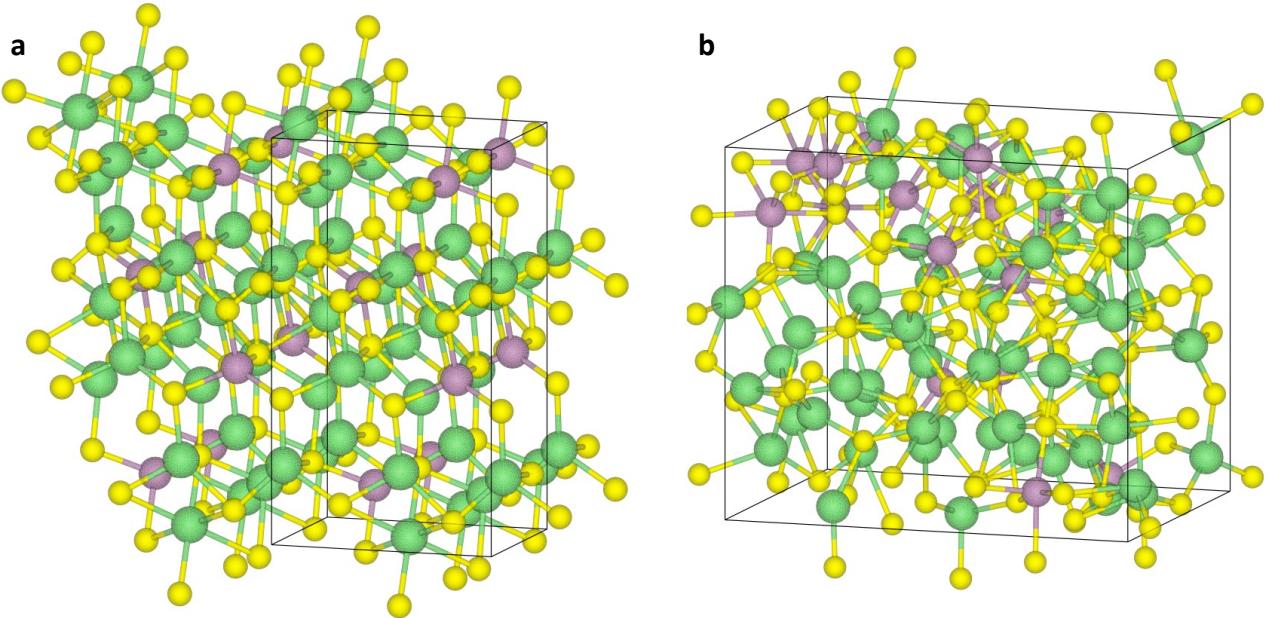


Figure Supplementary 8: **a** and **b** structures of Li_4MoS_4 before and after the solid phase transformation (separation of the Mo rich phase and the Li rich phase), respectively. Green: Li, yellow: S, purple: Mo.

from 1.22 kcal/mol/Å to 0.67 kcal/mol/Å, and RMSE of the virial tensor decreases from 164 kcal/mol to 87 kcal/mol. Detailed information about the training process of ensemble knowledge distillation is provided in the open-source code [23]. In Figure 2e in the main text and Figure Supplementary 7c, we demonstrate that ensemble knowledge distillation can significantly reduce the variation of density from MD simulations by almost 50%. Since density (ρ) correlates with the radial distribution function (RDF) as in equation (S53) where M^{avg} is the average atomic mass of the system, V_{total} is the total volume, and r is the radius of the sphere to calculate RDF:

$$\rho = \frac{M^{\text{avg}}}{V_{\text{total}}} * \int_0^\infty \text{RDF}(r) 4\pi r^2 dr = M^{\text{avg}} \langle \text{RDF}(r) \rangle, \quad (\text{S53})$$

the reduction of density fluctuation presented in this work also suggests the reduction of fluctuation of RDF. Therefore, the metric of density in this work is similar to the idea of using RDF to evaluate the fluctuation of MD in Fu *et al.* [31]. This illustrates the efficacy of ensemble knowledge distillation in reducing the fluctuation of MD results for liquids.

2.3 Case Study of Solid Phase Transformation

To demonstrate the broad applicability of ensemble knowledge distillation in improving the prediction performance and reducing fluctuations of results from MLFFs, we present another case study focusing on solid-phase transformation. Specifically, we investigate the system of Li_4MoS_4 , a composition similar to $\text{Li}_4\text{Mo}_{0.5}\text{Ti}_{0.5}\text{S}_4$, which is a potential cathode material for lithium-ion batteries [33]. In this study, we explore the distribution of Mo atoms within the system. As depicted in Figure Supplementary 8a, we begin by substituting Ti atoms in the crystal structure of Li_4TiS_4 (mp-766540 in the Materials Project database [34], with the space group of

Pnma) with Mo to create the Li_4MoS_4 phase with evenly distributed Mo atoms. We subsequently optimize the crystal structure and conduct 50 ps NPT ab-initio molecular dynamics (AIMD) simulations on the initial structure. As shown in Table [Supplementary 4](#), our results indicate that while the Mo atoms remain evenly distributed at 700 K, they begin to accumulate in the upper-left corner of the simulation box at 800 K and 900 K, as illustrated in Figure [Supplementary 8b](#). This accumulation suggests a phase transformation, specifically a phase separation between the Mo-rich phase and the Li-rich phase, during the AIMD simulations at 800 K and 900 K. The different behaviors at different temperatures might be related to the energy barrier associated with the phase transformation: the energy barrier might have the magnitude such that the thermal fluctuation at 700 K is hard to push the system through the barrier during the 50 ps MD simulation, while the thermal fluctuations at 800 K and 900 K are large enough to activate the system over the energy barrier.

In this case study, our objective is to utilize MLFFs to replicate the phase transformation and demonstrate how ensemble knowledge distillation can lower the fluctuation of reproducing the phase transformation. For this purpose, we select M3GNet, the first and one of the most widely used pretrained universal MLFFs for solids [35]. We perform 20 ps NPT AIMD simulations with a time step of 1 fs at 300 K, 1000 K, and 10 ps at 2000 K. The initial structure for all MD simulations consists of a $2 \times 2 \times 1$ supercell of crystalline Li_4MoS_4 with one random Li vacancy (totaling 143 atoms). Geometry optimization and AIMD are conducted using the Quantum Espresso package [36] with the PBE functional [37]. We utilize the first 10 ps trajectories of AIMD at 300 K and 1000 K as the training data to finetune the pretrained M3GNet, while the remaining 10 ps trajectories at 300 K, 1000 K, and 2000 K are used as the validation set. Employing stratified sampling with a frequency of 10 fs, we collect forces from a total of 2000 frames in AIMD trajectories as training data and 3000 frames as validation data. We optimize the pretrained M3GNet using 30 epochs with default hyperparameters as provided in the open-source code [35]. Here, we finetune the pretrained M3GNet with 10 different random seeds to form an ensemble of M3GNets. Using this ensemble, we predict forces for the 3000 frames in the validation set and further finetune the 10 M3GNets in the ensemble using the mean predicted forces of the validation set in conjunction with the DFT forces of the training set. Finally, we employ all M3GNets to conduct 500 ps NPT MD simulations to observe whether phase transformation occurs. Since the Mo atoms in the initial structure are separated by 5.9 Å from each other, we classify whether phase separation occurs by examining whether there is a pair of Mo atoms within 3 Å of each other in the last frame of the simulation.

In Table [Supplementary 3](#), the data presented includes the average force error for the initial M3GNets, the distilled M3GNets' average force error, and the force error from the mean predictions of the initial M3GNet ensemble in comparison with DFT across the validation set frames. It is evident that across all three temperatures, the distilled M3GNets exhibit a reduced average error compared to the initial M3GNets, yet this error exceeds that of the ensemble's mean predictions. This finding aligns with our prior analysis, suggesting that by diminishing $\delta_{\text{random}}^{\text{initial}}$ and minimizing $\delta_{\text{systematic}}^{\text{distill}}$ and $\delta_{\text{random}}^{\text{distill}}$ as in equation (S51), we can achieve $\|\vec{f}_{\text{pred}}^{\text{initial}} - \vec{f}_{\text{true}}\| < \|\vec{f}_{\text{pred}}^{\text{distilled}} - \vec{f}_{\text{true}}\| < \|\vec{f}_{\text{pred}}^{\text{initial}} - \vec{f}_{\text{true}}\|$. Moreover, the distilled M3GNets exhibit a significantly lower standard deviation in mean absolute error (MAE) compared to the initial M3GNets, indicating that employing ensemble knowledge distillation enhances the prediction error stability of M3GNets.

Table Supplementary 3: Mean absolute error (MAE) of prediction of forces from M3GNets compared with DFT. Unit of force is eV/Å.

Temperature	300K	1000K	2000K
Average MAE of initial M3GNets	0.0749 ± 0.008	0.1339 ± 0.008	0.2149 ± 0.035
Average MAE of distilled M3GNets	$0.0745 \pm 2 \times 10^{-5}$	$0.1309 \pm 3 \times 10^{-5}$	0.2036 ± 0.0002
MAE of mean predictions of initial M3GNets	0.0740	0.1288	0.2031

Table Supplementary 4: Behaviors observed from AIMD, initial MLFFs, and distilled MLFFs under different temperatures for the system of Li₄MoS₄. “Even” refers to the state where Mo atoms are evenly distributed, and “accumulated” the state where Mo atoms are accumulated in a small region of the simulation box.

Temperature	AIMD, 50 ps	10 initial MLFFs, 500 ps	10 distilled MLFFs, 500 ps
700K	Even	Even:Accumulated = 10:0	Even:Accumulated = 10:0
800K	Accumulated	Even:Accumulated = 4:6	Even:Accumulated = 1:9
900K	Accumulated	Even:Accumulated = 0:10	Even:Accumulated = 0:10

Table [Supplementary 4](#) illustrates that while both initial and distilled M3GNets reliably predict whether phase transformation will occur at 700K and 900K, there is a discrepancy at 800K. Here, 60% of the initial M3GNet models predict a phase transformation, whereas the remaining 40% do not. In contrast, post-ensemble knowledge distillation, 90% of the distilled M3GNets anticipate a phase transformation, underscoring that distilled M3GNets offer a more unified and reliable prediction in MD simulations. It is important to acknowledge the inherent randomness in MD simulations; theoretically, with infinite simulation duration, all M3GNets should predict phase transformation. However, given that MD simulations are typically constrained by time, the limited simulation time results in the observed variability in simulation outcomes as shown in Table [Supplementary 4](#). The presented case studies, encompassing both solid phase transformation and liquid electrolytes, demonstrate that ensemble knowledge distillation is an effective strategy for enhancing the reliability and consistency of MLFFs across different systems.

It is important to note that, in this work, we primarily emphasize that the ensemble knowledge distillation algorithm can reduce the fluctuations of results obtained from MD simulations powered by MLFF, not that it could make unstable MD simulations more stable (here “stable” means robust against collapse due to the possible spurious reactions or nonphysical configurations in the simulations). To prevent the MLFF-based MD from collapse, one should carefully design the DFT datasets and loss function to make the potential energy surface physical, smooth, and conservative. Please refer to the section above for our efforts on making the BAMBOO MLFF stable.

3 Supplementary Note: Density Alignment

3.1 Theory

In this work, we propose a physics-inspired approach to align the density predictions from MD simulations to experimentally measured density of liquids. The alignment is based on the isothermal compressibility of liquids:

$$\beta = \frac{1}{\rho} \left(\frac{d\rho}{dP} \right)_T, \quad (\text{S54})$$

where β is the isothermal compressibility of a given liquid, ρ is the density of the liquid, and P is the pressure of the system. If we assume β of a condensed liquid does not depend on P when P is not extremely high, which is a reasonable assumption in thermodynamics [38], then we have:

$$\Delta \ln(\rho) = \beta \Delta P. \quad (\text{S55})$$

Equation (S55) suggests a linear correlation between $\Delta \ln(\rho)$ and ΔP . This relationship is substantiated in Figure [Supplementary 9a-m](#), where the linearity holds across all examined liquids in the density alignment, evidenced by R^2 scores equal to or exceeding 0.98. This linear correlation implies that knowing the experimental density of a liquid and its corresponding density from an MD simulation allows us to determine the $\Delta \ln(\rho)$ between the two. Consequently, we can ascertain the ΔP required to adjust in the MD simulation to align the densities with experimental observations. Moreover, pressure in MD simulation is related to forces:

$$P = \frac{N_{\text{atoms}} k_B T}{V} + \frac{1}{3V} \sum_{1 \leq i < j}^{N_{\text{atoms}}} \langle \vec{f}_{ij}, \vec{r}_{ij} \rangle, \quad (\text{S56})$$

where k_B is the Boltzmann constant, T is the temperature, and V is the volume of the simulation box. Therefore, we find a bridge between the macroscopic observable density ρ and microscopic quantity pairwise forces \vec{f}_{ij} :

$$\Delta \ln(\rho) = \beta \Delta P = \beta \left(\Delta \left(\frac{N_{\text{atoms}} k_B T}{V} \right) + \Delta \left(\frac{1}{3V} \sum_{1 \leq i < j}^{N_{\text{atoms}}} \langle \vec{f}_{ij}, \vec{r}_{ij} \rangle \right) \right). \quad (\text{S57})$$

Since \vec{f}_{ij} is predicted by neural networks, equation (S57) provides a way to calculate the derivative of density to weights of neural networks, which enables us to align BAMBOO with experimentally measured density.

3.2 Performance

As illustrated in Figure 1a in the main text, we employ the ensemble distilled model as the foundational base model for performing density alignment. This alignment process encompasses 7 molecular liquids and 6 solutions, including 3 with LiPF₆ and 3 with LiFSI. To initiate the alignment, we first calculate the coefficient β for each liquid. This is achieved by training the distilled BAMBOO to adapt to various ΔP s, utilizing the subsequent loss function:

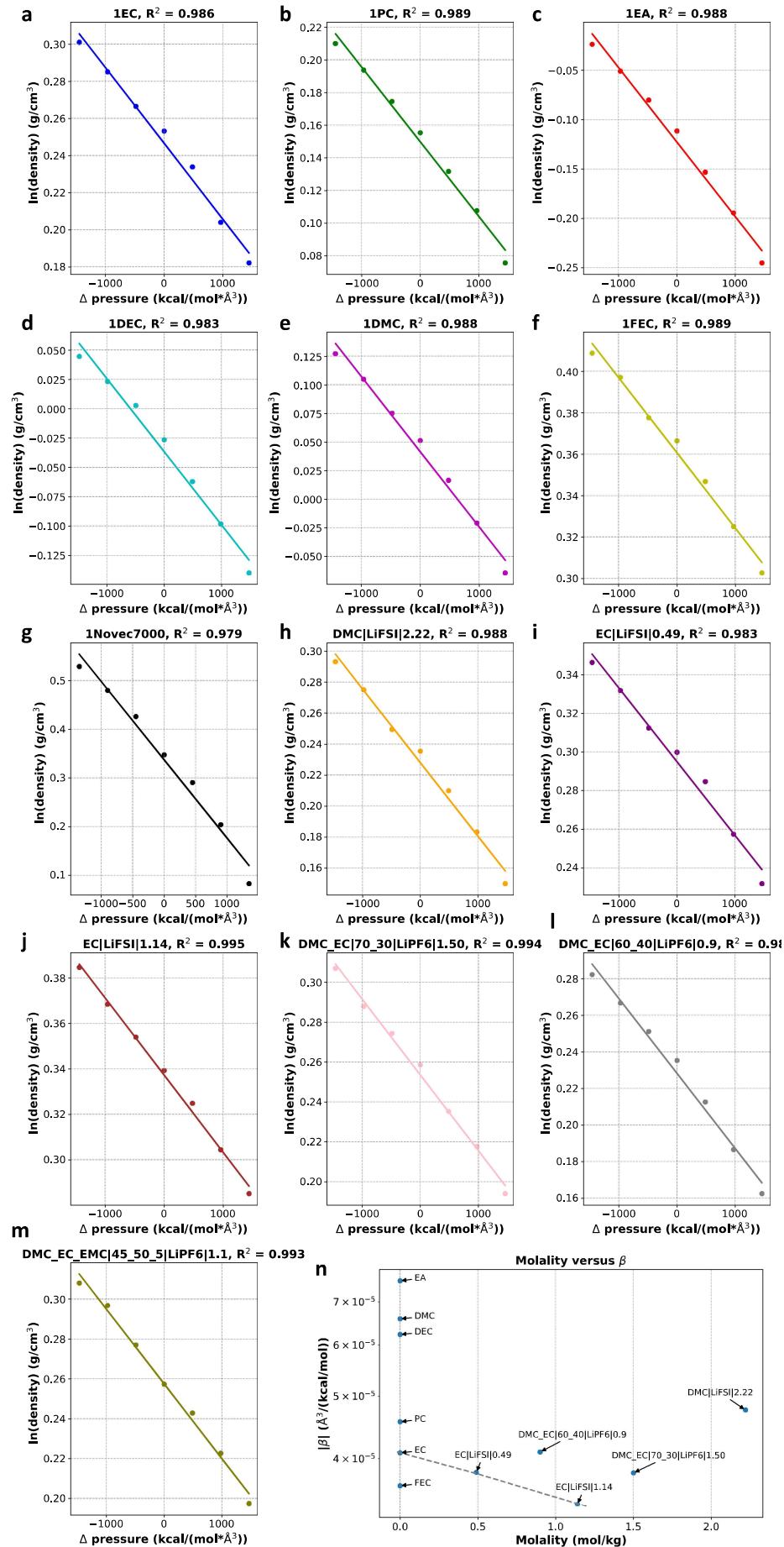


Figure Supplementary 9: **a** to **m** $\ln(\text{density})$ versus ΔP of the 13 systems included in density alignment, respectively. **n** Compressibility β derived from the linear relation versus weight of salts.

$$\begin{aligned}
L_{\text{align}} = L_{\text{training}} + \frac{1}{N_{\text{frame}}} \sum_k^{N_{\text{frame}}} & (\Delta P - \frac{1}{3V} \sum_{1 \leq i < j}^{N_{\text{atoms}}} \langle \vec{f}_{ij}^{\text{align}} - \vec{f}_{ij}^{\text{distill}}, \vec{r}_{ij} \rangle)^2 \\
& + \alpha_{\text{energy}}^{\text{align}} (E_k^{\text{align}} - E_k^{\text{distill}})^2 \\
& + \sum_i^{N_{\text{atoms}}} \alpha_{\text{force}}^{\text{align}} \|\vec{f}_{ki}^{\text{align}} - \vec{f}_{ki}^{\text{distill}}\|^2 \\
& + \alpha_{\text{virial}}^{\text{align}} \|\mathbf{T}_k^{\text{align}} - \mathbf{T}_k^{\text{distill}}\|^2.
\end{aligned} \tag{S58}$$

Here, ΔP is the assigned pressure change as in the x-axes in Figure [Supplementary 9a-m](#), $\sum \langle \vec{f}_{ij}^{\text{align}} - \vec{f}_{ij}^{\text{distill}}, \vec{r}_{ij} \rangle / (3V)$ is the pressure change of aligned MLFF compared with the base MLFF (we omit the $\Delta(N_{\text{atoms}} k_B T/V)$ term in equation (S57) as here we change pressure of the same frames without volume change, and the temperature change should also be negligible if N_{frame} is large enough). We also include the terms of $(E_k^{\text{align}} - E_k^{\text{distill}})^2$, $\|\vec{f}_{ki}^{\text{align}} - \vec{f}_{ki}^{\text{distill}}\|^2$, $\|\mathbf{T}_k^{\text{align}} - \mathbf{T}_k^{\text{distill}}\|^2$, and L_{training} to ensure that the aligned model would not drastically deviate from the initially trained model and the distilled model. In this work, we reuse the frames generated during the ensemble knowledge distillation for the alignment process as follows:

1. In the first round of alignment, we apply a uniform ΔP to each liquid for every model. This approach mirrors the ensemble knowledge distillation methodology, where we specifically adjust the weights of $\text{MLP}^{\text{energy}}$ within the MLFF for optimization. To enhance numerical stability, our focus is solely on the pressure changes stemming from inter-molecular interactions. Consequently, in equation (S58), we only include $\langle \vec{f}_{ij}^{\text{align}} - \vec{f}_{ij}^{\text{distill}}, \vec{r}_{ij} \rangle$, where atoms i and j do not belong to the same molecule or anion.
2. Following the initial alignment phase, we obtain a collection of aligned MLFFs, each corresponding to a distinct ΔP . Utilizing these differently aligned MLFFs, we conduct 600 ps NPT simulations to gather the resulting densities (ρ). Consequently, as depicted in Figure [Supplementary 9a-m](#), we establish a linear correlation between $\Delta \ln(\rho)$ and the assigned ΔP for each liquid. Leveraging this linearity, we can determine the β value for each liquid. This enables us to identify the specific ΔP that should be incorporated into the base MLFF for each liquid to align the $\Delta \ln(\rho)$ accurately.
3. In the final step, we employ the loss function specified in equation (S58), applying the unique ΔP identified for each liquid, to align the base MLFF with experimental data. This process finetunes the MLFF to ensure that its predictions for density closely match the experimental observations, thus enhancing the accuracy and reliability of the model in reflecting real-world phenomena.

In Figure [Supplementary 9n](#), we delve into the examination of the β values derived from the linear relationships. It is observed that molecules with similar structures exhibit comparable β values. For instance, cyclic carbonates like EC, PC, and FEC share similar β s, as do linear molecules such as DMC, DEC, and EA. Generally, there is a trend where higher viscosity correlates with lower compressibility, aligning with the intuition that liquids more resistant to shear stress are also more resistant to compression [39]. Additionally,

an increase in salt concentration leads to a decrease in β . For instance, in the case of EC and its solutions with LiFSI at different concentrations, there is a clear linear relationship with an R^2 score of 0.998, corroborating the empirical law that a solution's compressibility linearly decreases with an increase in salt molality (more salts result in less compressibility) [40]. The consistent compressibility trends, as deduced from density alignment, further confirm the validity of the density alignment approach.

Upon completing the second round of density alignment, the final MLFF is established. Figure [Supplementary 4](#) illustrates a slight decrease in the model's predictive accuracy for energy, forces, and virial compared with DFT calculations post-density alignment. Specifically, there is an 8% increase in the root mean square error (RMSE) for energy predictions, a 1% rise in forces RMSE, and a 4% uptick in virial RMSE. Despite these increases, the adjustments are made to enhance the model's accuracy in density predictions, which can be critical for certain applications. Detailed documentation of the training process for density alignment is accessible in the publicly available source code [23].

Figures [Supplementary 10](#) and 2 in the main text highlight the significant impact of density alignment on the physical properties of density, viscosity, and conductivity. The alignment process notably reduces the density error from 0.05 g/cm³ to approximately 0.01 g/cm³ for both the systems included in the alignment and those not directly aligned. This improvement suggests that density alignment, which adjusts the strength of inter-molecular interactions, has a degree of transferability to molecules with similar structures or solutions with varying salt concentrations not explicitly included in the alignment. For example, before alignment, cyclic carbonates like EC, FEC, and PC exhibit densities lower than experimental values. The alignment process increases inter-molecular interaction strength, thereby raising the predicted densities. Consequently, VC, another cyclic carbonate not included in the alignment, also shows an improved density prediction due to enhanced inter-molecular interactions among cyclic carbonates.

The effect of density alignment extends to solutions with EC and DMC as solvents and varying amounts of LiFSI salt. Prior to alignment, the MLFF tends to underestimate the densities of pure EC and DMC solvents. For aligned solutions at different molalities (0.49, 1.14, and 2.22 mol/kg), a pattern emerges: higher molality correlates with a lesser extent of density underestimation. This indicates that, initially, the MLFF underestimates solvent-solvent interactions while overestimating solvent-ion and/or ion-ion interactions. Through density alignment, these interactions are recalibrated—strengthening solvent-solvent and weakening solvent-ion/ion-ion interactions—leading to more accurate density predictions. Thus, for solutions with even higher molalities (3.70, 3.74, and 3.78 mol/kg) not included in the alignment, where the pre-alignment MLFF overestimated densities, the post-alignment MLFF, with adjusted interaction strengths, offers more accurate density predictions with a smaller degree of (or even no) overestimation.

In Figure [Supplementary 11](#), the observed strong dependence of viscosity and ionic conductivity on density aligns with expectations: a denser liquid typically exhibits stronger inter-molecular interactions, resulting in higher viscosity and lower conductivity. Thus, a post-alignment increase in density generally leads to a corresponding increase in viscosity and a decrease in conductivity. Figure [Supplementary 10b](#) and c confirm this relationship for most liquids post-alignment. From Figure [Supplementary 10b](#) and Figure 2g in the main text,

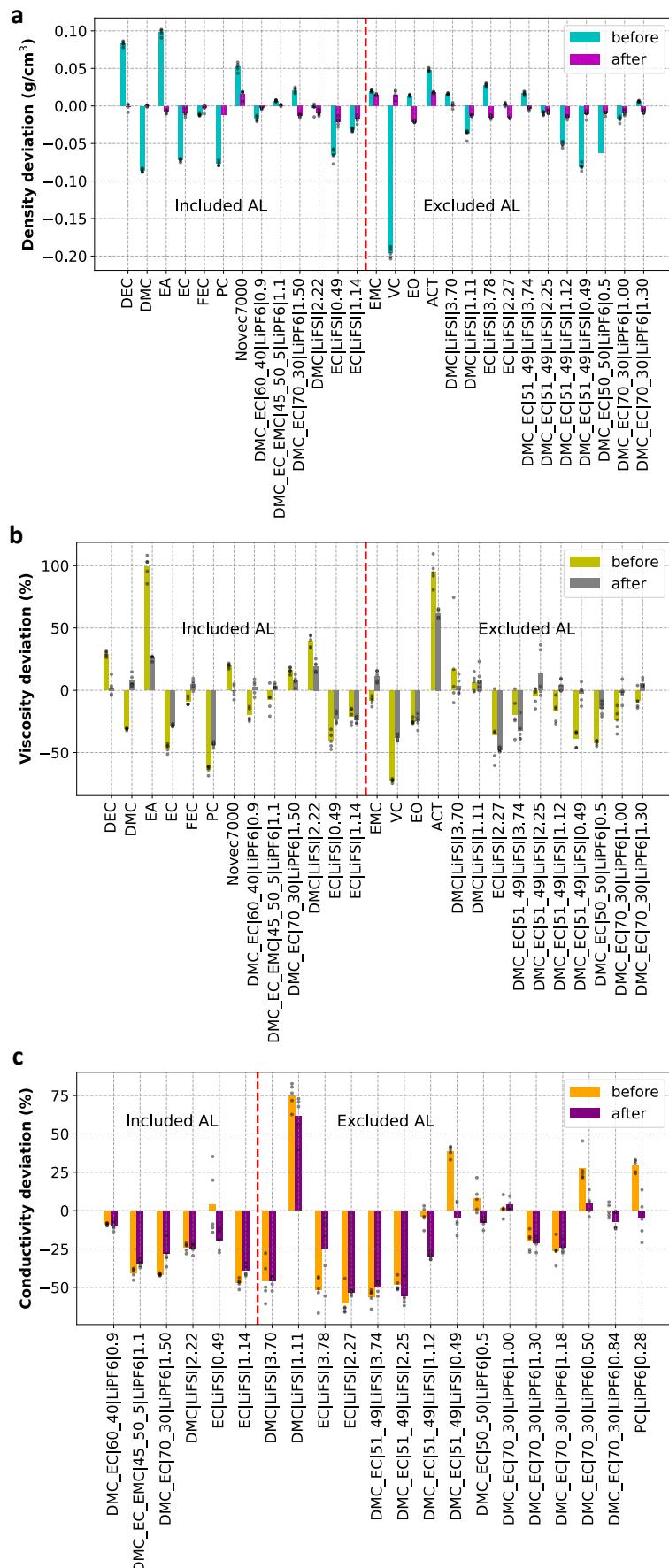


Figure Supplementary 10: **a** to **c** Density, viscosity, and conductivity of molecular liquid and solutions before and after density alignment, respectively. Bar plots reflect the mean values, and overlaid dots are values from 5 MD simulations with different random seeds.

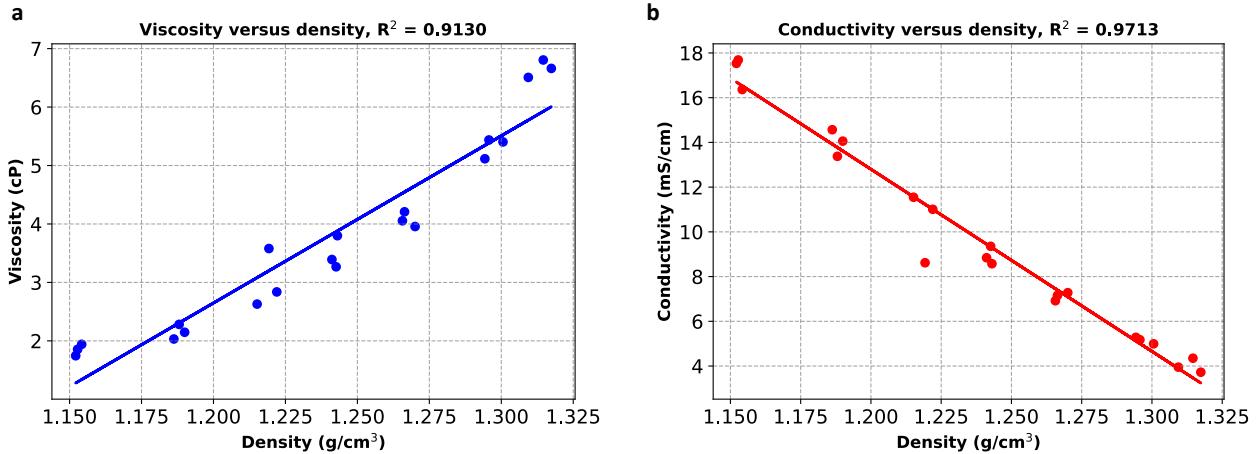


Figure Supplementary 11: **a** and **b** Density versus viscosity, and density versus conductivity for the system of DMC_EC|70_30|LiPF6|1.0, respectively. The models leading to different densities are from the first round of density alignment.

it's evident that the discrepancy in viscosity between MLFF MD simulations and experimental observations narrows significantly—from a 32% to a 17% deviation post-alignment. This improvement in viscosity accuracy is also seen in liquids not in the alignment, indicating the transferability of the alignment's effects.

However, the situation is more complex with ionic conductivity. In certain cases, aligning MD density to experimental data inadvertently moves the ionic conductivity further from the experimental values. For instance, the systems EC|LiFSI|0.49 and DMC_EC|0.51_0.49|LiFSI|1.12 show underestimations in density and viscosity but have conductivities close to experimental values before alignment. Post-alignment, while their densities and viscosities adjust closer to experimental figures, their conductivities decrease, diverging from the experimental data. Consequently, the overall deviation in ionic conductivity decreases only modestly from 31% to 26% after density alignment. This suggests that while density alignment effectively narrows the gap in density and viscosity predictions, a more targeted alignment strategy might be needed to align ionic conductivity predictions more closely with experimental results. In Section 7, we provide a more in-depth analysis about the predictions of individual systems of interest.

In Wang *et al.* [41], the gradient $\partial O / \partial \theta$, where O denotes a observable from MD simulations and θ parameters of force fields, is computed by the differentiable MD platform (DMFF) to optimize the observable from MD simulations. In this sense, both DMFF and the density alignment algorithm in this work can be used to finetune the force field by experimentally measured observables. However, to the best of the authors' knowledge, DMFF has been mostly used to optimize classic force fields and less applied to train deep learning-based MLFF. There are two possible reasons for this phenomenon. The first is that, DMFF requires multiple sequential MD simulations by the force fields to iteratively align the force field and the experimentally measured observables. For example, in Wang *et al.*[41], 40 to 60 sequential loops of MD simulations are conducted to align the radial distribution function (RDF) of one system. Since the inference speed of MLFF is still limited, DMFF and similar optimization methods are very expensive and time-consuming to optimize MLFF in practice. Instead, in the density alignment algorithm, for each system only one round of parallel MD simulations (7 in this work)

are conducted to compute the required change of pressure to align the MLFF and the experiment, which makes the optimization more affordable in terms of computational cost. The other possible reason is that, experimental datasets typically contain limited number of data points, which might make the MLFF overfit to the training data by the purely data-driven gradient-based optimization [42]. Since the density alignment algorithm is based on the compressibility of liquids and only optimizes inter-molecular interactions, the alignment demonstrates considerable transferability to liquids not initially included in the alignment process.

Despite the improved prediction of macroscopic properties, in future studies we will try to minimize the increase of error with respect to DFT during the alignment process. Here, we argue that the key to filling the gap between the reproduction of the experiment and DFT is to use the DFT method that is more in line with the experiment to generate the training data for the force field. An example is the thermal conductivity of water calculated by an MLFF trained with GGA or meta-GGA-level DFT data [43]. When trained on GGA generated data, the MLFF results in 60% overestimation of thermal conductivity; after switching to training data by the higher-level meta-GGA DFT functional, the deviation of thermal conductivity from the MLFF decreases to around 30% [43]. The dramatic influence of the level of DFT training data on the accuracy of macroscopic properties from the MLFF implies that one should carefully choose the DFT methods to train the force field, and it is necessary to benchmark the effect of different DFT training data on macroscopic properties from the trained MLFFs.

4 Supplementary Note: Ablation Study of Model Architecture

In this section, we provide the detailed results of the ablation study in Table [Supplementary 5](#), and description of the variants of GET (GIT, GE, and GET-no-charge). For GIT, we simply replace the equivariant relative vector $r_{ij} = (x_{ij}, y_{ij}, z_{ij})$ with an invariant feature $(\|r_{ij}\|, \|r_{ij}\|, \|r_{ij}\|)$, which keeps the model architecture. For GE, we skip the transformer architecture as in equation [\(S10\)](#) and equation [\(S11\)](#), and compute y_i^n as below:

$$y_i^n = \sum_j v_j^n \odot d_{ij}. \quad (\text{S59})$$

In the process of modifying the GET model, when the transformer architecture in the GE is omitted, the dimension of x_i^n within GE is increased from 64 to 76. Regarding the GET-no-charge variant, all components associated with the charge q_i are eliminated. This includes removing the MLPs that interact with q_i . To compensate for the reduction in complexity and to maintain a similar parameter count as the original GET, the number of layers in the $\text{MLP}^{\text{energy}}$ is increased from 2 to 4. These adjustments ensure that the modified models retain a comparable level of complexity, parameter count, and inference speed to the standard GET.

As shown in Table [Supplementary 5](#), GET exhibits superior predictive efficacy for both energy and forces derived from DFT when compared with GE and GIT. This observation intimates that equivariance might play a pivotal role, surpassing that of the transformer, as evidenced by GE's reduced error margins relative to GIT. In the case of GET-no-charge, a significantly elevated error in forecasting DFT energy is noted in comparison

Table Supplementary 5: Results of ablation study of models trained on DFT data without ensemble knowledge distillation and density alignment. Here, “number of parameters” is the total number of degree of freedom of each MLFF that can be optimized in the training. “Density MAE” is based on the 13 systems included in density alignment as in Figure [Supplementary 10](#). Units: Energy: kcal/(mol*atom); Forces: kcal/(mol*Å); Virial: kcal/mol; density: g/cm³

	GET	GIT	GE	GET-no-charge
Number of parameters	103,620	103,620	108,900	107,649
Energy RMSE	0.245±0.003	0.278±0.002	0.269±0.003	0.812±0.003
Forces RMSE	2.19±0.02	2.96±0.02	2.32±0.01	2.01±0.02
Virial RMSE	7.54±0.05	9.41±0.09	7.95±0.05	7.69±0.09
Density MAE before distillation	0.049±0.007	0.114±0.028	0.065±0.020	0.082±0.033

to GET. This discrepancy can be ascribed to the intrinsic long-range dependence characteristic of electrostatic energy, which is inversely proportional to distance ($1/r$), coupled with the relatively diminished capacity of GNN to encapsulate long-range interactions as opposed to short-range dynamics [44]. Pertaining to the prediction of forces, GET-no-charge registers a marginally lower error than GET. This anomaly could potentially be attributed to the inherent limitations associated with the CHELPG method of labelling partial charges [17]. While electrostatic force exhibits lower sensitivity to long-range information compared with energy ($1/r^2$ for force and $1/r$ for energy), CHELPG is calibrated against the electrostatic potential at a considerable distance away from the cluster. Consequently, CHELPG is not optimally designed to capture short-range electrostatic forces, and it is numerically unstable as discussed above. The employment of CHELPG for computing short-range electrostatic forces may inadvertently introduce non-physical biases into the neural network (NN)-facilitated force learning process, thereby exacerbating the learning challenge for the NN with an increase in error. This finding underscores the imperative for the development of more advanced abstract representations of electron density that can accurately and efficiently compute electrostatic forces. Finally, GET is distinguished by its minimal error in modeling density from MD simulations in contrast to GE, GIT, and GET-no-charge, a testament to its overarching superior capability in the precise prediction of DFT-calculated quantities.

To further investigate the difference between GET and GE after density alignment, we perform the same density alignment operation on the GE model as that of the GET model. After alignment, the density MAE of GE decreases from 0.065 g/cm³ to 0.018 g/cm³ for systems in Table S5. Although the density alignment algorithm can significantly decrease the density MAE of GE by 72%, the MAE of 0.018 g/cm³ of GE is still much higher than that of GET after alignment (0.011 g/cm³). Here, we attribute the difference to the initial density error and the convergence of model finetuning. In density alignment, we use the target pressure change ΔP to align density from MD and experiment. According to equation (S55), larger density difference $\Delta \ln(\rho)$, larger ΔP . Since GET has smaller density MAE than GE before alignment, the mean absolute ΔP for GET (477 kcal/(mol*Å³)) is also smaller than that of GE (832 kcal/(mol*Å³)). In density alignment, we try to simultaneously change the pressure from MLFF, and confine the MLFF to still approximate DFT by including the initial DFT training loss L_{training} in the finetuning loss as in equation (S58). Intuitively, if the target pressure change ΔP becomes larger, then the finetune process will be harder to converge. After finetune, GE has a larger

difference between the target pressure change and the actual change than GET (80 kcal/(mol*Å³) versus 56 kcal/(mol*Å³)). Moreover, for GE there is an 28% energy prediction RMSE increase compared with DFT, 1% force RMSE increase, and 8% virial RMSE increase. Compared with that of GET (8% energy RMSE increase, 1% forces RMSE increase, and 4% virial RMSE increase), we can see that the density alignment algorithm deviates GE further away from DFT than GET. These two observations suggest that GE with higher initial density error is harder to be aligned than GET, which might explain why GE performs worse than GET after alignment. The comparison between GE and GET also suggests that density alignment cannot solve everything, and one should make the MLFF before density alignment as accurate as possible for better performance after alignment.

In addition to the significant modifications of model architecture (GIT, GE, and GET-no-charge), we also try GET models with different dimensions of node embedding. When the dimension is decreased to 32, the validation energy RMSE raises 32.3%, force RMSE 8.6%, and virial RMSE 15.6%, and when the dimension is increased to 128, the validation energy RMSE only decreases 2.8%, force RMSE 8.5%, and virial RMSE 5.9%. As a result, we choose the 64 dimensional node embedding in this work for the balance between prediction accuracy and cost. We suggest that, for tasks with higher computational demand, such as MD simulation of polymers, one could choose the GET architecture with smaller model sizes or simplified models such as GE, while for tasks where computational cost is not considerable, such as direct prediction of molecular/materials properties, one could consider the GET model with larger model sizes and cutoff radius, or even more complicated architectures such as the L = 8 EquiformerV2 [45] model with millions of parameters.

As mentioned in the main text, the GET model in this work is inspired by TorchMD-Net [46]. In the following, we introduce the sequential stages of modifications of TorchMD-Net to the final GET model. The first modification of GET in this work compared with TorchMD-Net is the way to compute attention scores. In this work, we connect atoms (nodes in the graph) within the cutoff radius, and when we compute the attention scores a_{ij}^n , we only consider atom pairs that are connected by edges in the graph. Therefore, we claim that we use an “attention mechanism on graph” such that the attention mechanism is applied only on pairs of nodes that are connected in the graph. In TorchMD-Net [46], the attention scores are computed for all pairs of atoms in the graph. Therefore, we claim that TorchMD-Net uses the “global attention” that computes attentions of all pairs of nodes as in the original transformer architecture [5]. The main reason for us to use “attention mechanism on graph” instead of “global attention” is the scaling of computational cost with respect to number of atoms in the graph: for “attention mechanism on graph” the scaling is $\mathcal{O}(N)$, while for “global attention” the scaling is $\mathcal{O}(N^2)$. The reason behind the use of “global attention” in TorchMD-Net might be that it was mainly used in the corresponding paper [46] to predict properties of single molecules where the inference speed is not largely affected by the $\mathcal{O}(N^2)$ scaling, and the reason for us to choose the “attention mechanism on graph” is that we primarily use the GET model to perform MD simulations with thousands of atoms, where the $\mathcal{O}(N^2)$ scaling is not affordable.

Moreover, in this work we use an intermediate vector representation \vec{u}_i^n to incorporate edge vector representation as in equation (S12). In TorchMD-Net, \vec{u}_i^n is constructed as below:

$$\vec{u}_i^n = \sum_j v_j^n * \vec{e}_{ij} + \sum_j v_j^{n,\text{res}} * \vec{V}_j^{n-1}. \quad (\text{S60})$$

Comparing equation (S12) and (S60), we can see that equation (S60) has an additional operation $\sum_j v_j^{n,\text{res}} * \vec{V}_j^{n-1}$, which is essentially a residual connection that adds the node vector representation of the previous layer \vec{V}_j^{n-1} to the intermediate vector representation \vec{u}_i^n . This operation is time consuming as it requires additional neural network parameters to generate the second “value” tensor $v_j^{n,\text{res}}$ from the node scalar representation x_i^n , and it sums over all edges in the graph. On the other hand, since the main role of \vec{u}_i^n is to introduce the equivariant edge information into the network, we think that the second term in equation (S60) might downplay the importance of the first term, similar to the situation of over-squashing of GNN [47]. After we replace equation (S60) with (S12), we find that the inference speed increases 24% for systems with 10,000 atoms, and the energy prediction RMSE decreases 2.9%, force RMSE increases 4.5%, and virial RMSE increases 6.9%. Therefore, we use equation (S12) to construct \vec{u}_i^n in the GET model in this work to achieve a noticeable acceleration with a moderate drop of prediction accuracy.

Another modification is about the construction of the edge scalar representation d_{ij} . In GET, d_{ij} is computed as in equation (S4), while in TorchMD-Net the edge scalar representation is computed separately in each layer as in equation (S61):

$$d_{ij}^n = \text{SiLU}(W^{d,n} \text{RBF}(\cos \frac{\pi \|r_{ij}\|}{r_{\text{cut}}} + 1)). \quad (\text{S61})$$

Since the input to compute d_{ij}^n is the same ($\text{RBF}(\cos \frac{\pi \|r_{ij}\|}{r_{\text{cut}}} + 1)$) for each layer n , we think that using a single d_{ij} for all layers instead of multiple $\{d_{ij}^n\}$ s might not significantly degrade prediction performance while achieves considerable acceleration. After replacing equation (S61) with (S4), we find that the inference speed increases 13.1% for 10,000 atoms while the energy prediction RMSE decreases 0.9%, force RMSE increases 1.6%, and virial RMSE increases 4.1%. Therefore, we use equation (S4) in the GET model.

The last change is about the input of distance into computing the attention score a_{ij}^n . In GET, the interatomic distance $\|r_{ij}\|$ is directly used to compute the attention score as in equation (S10), while in TorchMD-Net, another edge scalar representation d_{ij}^a is constructed as in equation (S62) to compute the attention score as in equation (S63):

$$d_{ij}^a = \text{SiLU}(W^{d,a} \text{RBF}(\cos \frac{\pi \|r_{ij}\|}{r_{\text{cut}}} + 1)), \quad (\text{S62})$$

$$a_{ij}^n = \text{SiLU}(\langle q_i^n, k_j^n \odot d_{ij}^a \rangle). \quad (\text{S63})$$

Here, we think that the introduction of $W^{d,a}$ could not largely increase the capacity of the network, because on the one hand, $W^{d,a}$ is right before two consecutive SiLU [3] non-linear activation functions without any residual connection, which could cause the effect of vanishing gradient [48] to $W^{d,a}$ and make the network parameters

hard to optimize. On the other hand, when computing the intermediate node scalar representation y_i^n as in equation (S11), W^d already serves the role of transforming distance information to compute y_i^n . After we replace equation (S63) with (S10), we find that the inference speed increases 18% for systems with 10,000 atoms, and the energy prediction RMSE decreases 1.4%, force RMSE decreases 2.6%, and virial RMSE decreases 3.1%. Therefore, we switch to equation (S10) to compute the attention score.

5 Supplementary Note: Details of Molecular Dynamics

We employ LAMMPS [13] to conduct the MD simulations for BAMBOO. For each system, an initial step involves energy minimization, which is constrained to a maximum of 1000 iterations and 100,000 evaluations. Following this, the liquid systems are subjected to equilibrium under the NPT ensemble for 1 ns, with a step size maintained at 1 fs. The density values presented in this paper represent the average measured during the final 50 ps of the NPT simulation. Subsequently, we carry out a 4 ns production run under the NVT ensemble to determine the viscosity, diffusivity, and ionic conductivity of the systems.

The viscosity (η) and diffusivity (D) of the pure solvents and various mixtures are computed from equilibrium simulations by applying an Einstein relation as described below:

$$\eta = \frac{V}{2k_B T} \lim_{t \rightarrow \infty} \frac{d}{dt} \left\langle \left(\int_{t_0}^{t_0+t} dt' P_{\alpha\beta}(t') \right)^2 \right\rangle_{t_0}, \quad (\text{S64})$$

$$D = \frac{1}{6N} \lim_{t \rightarrow \infty} \frac{d}{dt} \left\langle |\vec{r}_{i,t+t_0} - \vec{r}_{i,t_0}|^2 \right\rangle_{i,t_0}, \quad (\text{S65})$$

where V and T represent the volume and the temperature of the simulations. $P_{\alpha\beta}(t')$ is the shear component of the stress tensor at the time step t' . N represents the number of ions or molecules at interest, and $\vec{r}_{i,t}$ the 3D coordinate of ion/molecule i at time t , t_0 the starting time step, and t the total number of time steps in the simulation trajectories used for viscosity or diffusivity analysis.

Following the calculation of diffusion coefficients, we calculate transference number using the following equation:

$$t_+ = \frac{D_+}{D_+ + D_-}. \quad (\text{S66})$$

Two approaches are used to calculate ionic conductivities, the Nernst-Einstein method and a method described by Mistry *et al* [49] that built on Stefan-Maxwell diffusivities. In this work, by default the ionic conductivity is calculated by Mistry *et al* [49] unless specified. The Nernst-Einstein equation is described below:

$$\sigma_{NE} = \frac{e^2}{V k_B T} (N_+ z_+^2 \bar{D}_+ + N_- z_-^2 \bar{D}_-), \quad (\text{S67})$$

where e is the elementary charge, k_B the Boltzmann constant, and V and T are volume and temperature of the simulated system. \bar{D}_\pm , z_\pm , and N_\pm are the self-diffusion coefficients, the charges, and the counts of the positive and negative ions, respectively.

In our calculations for the diffusion coefficients and conductivities, we apply finite size correction factor (FSC) to account for the effect of simulation box sizes [50]. The FSC is calculated using the following equation:

$$\text{FSC} = \frac{2.837298 \times k_B \times T}{6\pi\eta L}, \quad (\text{S68})$$

where η is the estimated viscosity and L is the simulation box length.

6 Supplementary Note: Comparison with Other Force Fields

In this section, we compare the performance between BAMBOO and APPLE&P [51], QRNN [30], and OPLS-AA [52, 53, 54] as in Table [Supplementary 6](#), [Supplementary 7](#), and [Supplementary 8](#). Here, the properties from APPLE&P and QRNN are collected from the corresponding references listed in the tables, and those from OPLS-AA are computed by ourselves. For OPLS-AA, the force field parameters for solvent molecules are taken from Ref. [52], that for Li from Ref. [53], and that for anions from Ref. [54]. Here, the system size of OPLS-AA-based MD simulation increases from 4,000 atoms (the size for BAMBOO MLFF) to 10,000 atoms, and all other settings for the OPLS-AA simulations are the same as that for BAMBOO-based simulations.

From Table [Supplementary 6](#), we can see that, for both density of pure EC, DMC, PC, and conductivities of liquid electrolytes with DMC and EC as solvents and LiPF₆ or LiTFSI as salts, BAMBOO predicts more closely to the experimental values than APPLE&P. We can also see that, for systems in Table [Supplementary 8](#) and [Supplementary 9](#) which are included in the DFT dataset, BAMBOO achieves lower prediction error for density (MAE 0.011 g/cm³), viscosity (mean deviation 17.3%), and conductivity (mean deviation 24.2%) than OPLS-AA (0.020 g/cm³, 140.4%, and 54.9% for density, viscosity, and conductivity error, respectively). The two observations show the superior ability of BAMBOO on simulating bulk properties of organic liquids and electrolytes compared with classic force fields with and without polarizable terms. In Table [Supplementary 7](#), we compare density and viscosity from BAMBOO and QRNN, from which we can see that BAMBOO outperforms QRNN in terms of density and viscosity prediction, which demonstrates the state-of-the-art performance of BAMBOO among existing machine learning force fields.

7 Supplementary Note: Properties of Simple Liquid Electrolytes

In Figure [Supplementary 12](#), we present the predicted physical properties of simple solvents and electrolytes using BAMBOO compared with available experimental measurements. Figure [Supplementary 12a](#) displays a parity plot between the experimental and predicted densities. To differentiate the alignment treatment, we use red and green markers in Figure [Supplementary 12a](#) to indicate the systems that are included and excluded

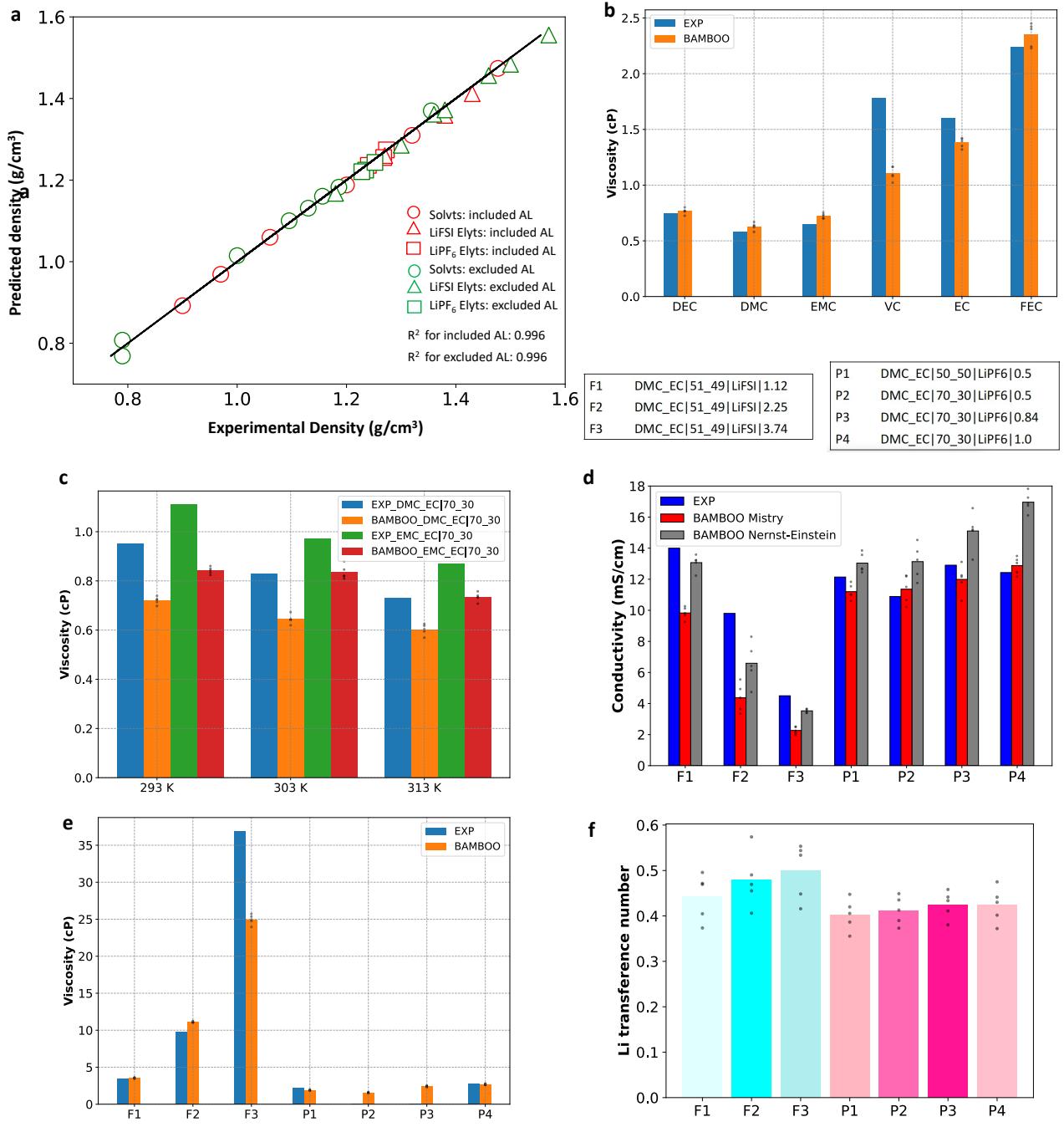


Figure Supplementary 12: The predicted solvents and liquid electrolytes' properties using BAMBOO.

a. The parity plot comparing the predicted and experimentally measured densities of pure solvents, solvent mixtures, LiPF₆ and LiFSI containing electrolytes. “AL” in the legend represents the density “alignment” treatment.

b. The predicted viscosity of common solvents compared to experimental measurements.

c. Comparing the predicted viscosity of DMC-EC and EMC-EC solvent mixtures as a function of temperature to experimental results.

d. Comparing the predicted and experimental conductivity of 7 common liquid electrolytes. The conductivity values presented are obtained by analysing the simulated trajectories using Mistry's method [49] and Nernst-Einstein equation, denoted as “BAMBOO Mistry” and “BAMBOO Nernst-Einstein”, respectively.

e. The predicted and experimental viscosity of the 7 electrolytes presented.

f. The predicted cation transference number calculated using self-diffusion coefficients of Li⁺ and anions of the 7 electrolytes. The tables below panels d-f describe the electrolyte composition in “solvents|solvent weight %|salt|salt molality” format. All data are tabulated in Table [Supplementary 12](#) to [Supplementary 16](#). In all the figures, bar plots reflect the mean values, and overlaid dots are values from 5 MD simulations with different random seeds.

Table Supplementary 6: The simulated density (ρ , in g/cm³), and ionic conductivity (σ , in mS/cm) of solvents and liquid electrolytes from APPLP&P [51] and BAMBOO. For conductivities reported in the reference that have multiple records from different post-processing methods, the value closest to the experimental is shown here, so as BAMBOO here for a fair comparison. The uncertainty for the density calculated using BAMBOO ranges from 0.001 to 0.005 g/cm³. The uncertainty for the conductivities calculated using BAMBOO ranges from 0.089 to 0.425 mS/cm. Except the last row where the temperature is specified as 333 K, all other BAMBOO simulations are conducted at 298 K. For clarity, we do not show simulated properties which do not have corresponding experimental/simulated values to compare.

Compositions	$\rho_{\text{APPLP\&P}}$	ρ_{BAMBOO}	ρ_{Exp}	$\sigma_{\text{APPLP\&P}}$	σ_{BAMBOO}	σ_{Exp}
DMC	1.040[51]	1.060	1.057[51]	NA	NA	NA
EC	1.302[51]	1.310	1.321[51]	NA	NA	NA
PC	1.209[51]	1.188	1.194[51]	NA	NA	NA
DMC_EC 51_49 LiPF ₆ 0.86	NA	NA	NA	13.5[55]	11.740	12[55]
DMC_EC 51_49 LiTFSI 0.86	NA	NA	NA	12[55]	6.359	9[55]
DMC_EC 51_49 LiPF ₆ 1.12	NA	NA	NA	15.9[56]	10.269	11.7[57]
The formula above at 333K	NA	NA	NA	25.9[56]	20.490	19.5[57]

Table Supplementary 7: The simulated density (ρ , in g/cm³), and viscosity (η , in cP) of solvents and liquid electrolytes from QRNN [30] and BAMBOO. The uncertainty for the density calculated using BAMBOO ranges from 0.001 to 0.005 g/cm³. The uncertainty for the viscosity calculated using BAMBOO ranges from 0.018 to 0.275 cP, while that from QRNN ranges from 0.05 to 3.77 cP. The BAMBOO simulations for pure solvents are conducted at 298 K, and that for DMC, EC, LiPF₆ mixtures are conducted at 313 K. For clarity, we do not show simulated properties which do not have corresponding experimental/simulated values to compare.

Compositions	ρ_{QRNN}	ρ_{BAMBOO}	ρ_{Exp}	η_{QRNN}	η_{BAMBOO}	η_{Exp}
DMC	1.0[30]	1.060	1.057[51]	0.79[30]	0.629	0.585,0.706[58, 59]
EC	1.35[30]	1.310	1.321[51]	2.27[30]	1.381	1.60-1.90[58, 60, 59]
PC	1.22[30]	1.188	1.194[51]	2.6[30]	1.426	2.53[58]
EMC	0.99[30]	1.015	1.0[58]	0.65[30]	0.724	0.65[58]
DEC	0.92[30]	0.969	0.97[58]	0.43[30]	0.766	0.749[58]
VC	1.31[30]	1.370	1.355[58]	1.26[30]	1.104	1.5,1.78[30, 61]
FEC	1.47[30]	1.474	1.477[60]	3.02[30]	2.349	2.24-4.1[60, 62, 59]
DMC_EC 70_30	NA	NA	NA	0.83[30]	0.756	0.73[63]
DMC_EC 70_30 LiPF ₆ 0.51	NA	NA	NA	1.65[30]	1.448	1.18[63]
DMC_EC 70_30 LiPF ₆ 1.02	NA	NA	NA	2.02[30]	2.398	1.92[63]
DMC_EC 70_30 LiPF ₆ 1.63	NA	NA	NA	6.98[30]	4.328	3.21[63]
DMC_EC 70_30 LiPF ₆ 2.14	NA	NA	NA	22.38[30]	6.351	5.1[63]

from the density alignment process, respectively. Despite the different treatment, both sets of data exhibit high R² values of 0.996, indicating excellent agreement between the predicted and experimental values. We observe that the densities of most linear solvents cluster below 1.2 g/cm³, whereas those of cyclic solvent species typically register higher values. Notably, the presence of the “C-F” functional group leads to an increase in solvent density. This density trend by solvent category follows chemical intuition. The densities of LiPF₆-based electrolytes aggregate between 1.2 g/cm³ to 1.3 g/cm³, reflecting the consistent presence of roughly 1m salt (molality, in mol/kg) in these electrolytes for which experimental values are available. In contrast, experimental records of LiFSI-based electrolytes span a wider concentration range, from 0.49m to 3.74m, resulting in a broader spread of densities from 1.2 g/cm³ to 1.6 g/cm³. Furthermore, our model predicts the densities of EO (CH₃CH₂OH) and ACT (CH₃-CO-CH₃) with good accuracy despite their structural differences from those included in density alignment. This results may be attributed to the transferability of the alignment process

Table Supplementary 8: The simulated density (ρ , in g/cm³), viscosity (η , in cP), and ionic conductivity (σ , in mS/cm) of solvents and liquid electrolytes from OPLS-AA [52, 53, 54]. For simplicity, systems shown in Table [Supplementary 9](#), [Supplementary 10](#), [Supplementary 11](#) are not repeated here, and the experimental values shown in Tables listing BAMBOO predictions in section [11](#) are not repeated here. The uncertainty for the density calculated using OPLS-AA ranges from 0.001 to 0.004 g/cm³, that of viscosity from 0.002 to 9.972 cP, and that of conductivity from 0.037 to 0.736 mS/cm. For clarity, we do not show simulated properties which do not have corresponding experimental values to compare. All simulations are conducted at 298K unless specified.

Compositions	$\rho_{\text{OPLS-AA}}$	$\eta_{\text{OPLS-AA}}$	$\sigma_{\text{OPLS-AA}}$
DEC	0.995	1.146	NA
DMC	1.081	0.892	NA
EA	0.924	0.709	NA
EC	1.273	1.073	NA
FEC	1.458	1.264	NA
PC	1.194	1.426	NA
Novec7000	1.492	0.449	NA
DMC 60_40 LiPF ₆ 0.9	1.250	6.534	5.808
DMC EMC 45_50_5 LiPF ₆ 1.1	1.283	9.266	4.703
DMC EC 70_30 LiPF ₆ 1.5	1.283	15.899	3.294
DMC LiFSI 2.22	1.285	21.700	2.498
EC LiFSI 0.49	1.334	4.241	4.328
EC LiFSI 1.14	1.389	10.689	2.971
EMC	1.031	0.999	NA
VC	1.340	0.704	NA
EO	0.773	0.513	NA
ACT	0.788	0.375	NA
EP	0.898	0.699	NA
DMC LiFSI 3.70	1.382	138.552	0.489
DMC LiFSI 1.11	1.190	5.645	5.047
EC LiFSI 3.78	1.557	136.279	0.538
EC LiFSI 2.27	1.472	34.459	1.458
DMC EC 51_49 LiFSI 3.74	1.465	211.159	0.369
DMC EC 51_49 LiFSI 2.25	1.374	33.181	1.529
DMC EC 51_49 LiFSI 1.12	1.287	7.56	4.474
DMC EC 51_49 LiFSI 0.49	1.229	3.291	5.487
DMC EC 50_50 LiPF ₆ 0.5	1.231	3.851	5.621
DMC EC 70_30 LiPF ₆ 1.0	1.239	7.051	5.371
DMC EC 70_30 LiPF ₆ 1.3	1.267	11.429	4.018
DMC EC 70_30 LiPF ₆ 0.5	NA	NA	5.921
DMC EC 70_30 LiPF ₆ 0.84	NA	NA	7.887
PC LiPF ₆ 0.28	NA	NA	3.345
EC LiPF ₆ 0.87	NA	NA	5.606
DMC EC 51_49 LiPF ₆ 0.86	NA	NA	6.236
DMC EC 51_49 LiTFSI 0.86	NA	NA	5.133
DMC EC 51_49 LiPF ₆ 1.12	NA	NA	6.236
DMC EC 51_49 LiPF ₆ 1.12 313K	NA	NA	14.455
DMC EC 70_30 313K	NA	1.23[30]	NA
DMC EC 70_30 LiPF ₆ 0.51 313K	NA	2.15[30]	NA
DMC EC 70_30 LiPF ₆ 1.02 313K	NA	3.62[30]	NA
DMC EC 70_30 LiPF ₆ 1.63 313K	NA	7.80[30]	NA
DMC EC 70_30 LiPF ₆ 2.14 313K	NA	15.26[30]	NA

using ethyl acetate (EA, CH₃COOC₂H₅) and other linear solvent species, underscoring the transferability of the density alignment algorithm.

Figure [Supplementary 12b](#) and [Supplementary 12c](#) illustrate the predicted viscosity values for both pure

solvents and binary solvent mixtures, respectively. We specially selected these solvents to address the broader interests of the Li battery electrolyte community. In examining pure solvents, we observe good agreements between the predicted viscosity values for linear solvents (DEC, DMC, and EMC) and experimental measurements. However, some discrepancies are apparent for cyclic carbonates, particularly VC, where the MLFF underestimates its viscosity by 0.67 cP. It's important to acknowledge that experimental viscosity values reported in the literature for the same solvent can vary significantly. For instance, the reported viscosities for FEC at 40°C range from 2.24 cP to 4.1 cP [59, 60, 62]. Therefore, while discrepancies exist, it remains uncertain whether the performance of BAMBOO on some carbonates is accurately assessed using the available experimental results. Nevertheless, we can confidently assert that the differences in viscosity stemming from the categories of solvents are accurately captured by BAMBOO. Specifically, linear carbonates consistently exhibit lower viscosity compared with cyclic carbonates.

In Figure [Supplementary 12c](#), we showcase the predicted binary solvent viscosity (red markers) for DMC:EC 70:30 wt% and EMC:EC 70:30 wt% as a function of temperature, contrasted with experimental results (blue markers) reported by Logan *et al.* [63]. Notably, the decreasing trend depicted by the predicted values mirrors the experimental measurements. However, a consistent underestimation of viscosity by 0.2 cP is evident. Nonetheless, despite the underestimation, BAMBOO describes the trend such that EMC-EC viscosity exceeds DMC-EC.

Figure [Supplementary 12d](#) and [Supplementary 12e](#) show the conductivity and viscosity of 7 selected LiFSI- and LiPF₆-based electrolyte blends. Among these, the three LiFSI based electrolytes are selected to demonstrate the variations in properties with salt concentration. Meanwhile, the inclusion of four LiPF₆-based electrolytes aims to emphasize changes resulting from different solvents while maintaining a fixed salt concentration, as well as variations arising from different salt concentrations while keeping the solvent constant. This design is to emphasize trends and correlations in transport properties. It is known that conductivity and viscosity have strong correlation, such that, the lower the viscosity, the higher the conductivity. This correlation is also widely used in designing fast charging liquid electrolytes [64, 63, 65]. Two methods are used to calculate conductivity, a method described by Mistry *et al.* [49] that uses Stefan-Maxwell diffusivities, and the well-known Nernst-Einstein (NE) relation in equation [S67](#) that uses self-diffusivities of the positive and the negative ions and ignores ion-pairing and molecules' correlated motion in simulation. Due to the nature of diffusivity calculations, the conductivity via Mistry's method for a typical liquid electrolyte is always lower than the results from NE, but converges to NE at extremely low salt concentration. Therefore, the consistent lower results by Mistry's method manifest the fact that ion-pairing and correlated motion exist in the simulated electrolytes, consistent with our solvation structure analysis in Figure 3 in the main text.

In Figure [Supplementary 12d](#), moving from left to right, we observe a decrease in the experimental conductivity of LiFSI-based electrolytes (F1, F2, and F3) with increasing salt concentration. This trend is effectively captured by conductivities calculated using both methods; however, Mistry's method exhibits a larger underestimation compared to NE. The associated atomic charge distributions for these three electrolytes are presented in Figure 3 in the main text to illustrate charge evolution as a function of concentration. Regarding LiPF₆-based

electrolytes (P1-P4), Mistry's method yields considerably more accurate results than NE. When considering error bars, the predicted results align well with the experiments. In contrast, NE estimated-conductivities of P3 and P4 are much higher than P1 and P2, though the differences across all four blends (P1-P4) are not significant.

Figure [Supplementary 12e](#) shows the predicted viscosities of electrolyte blends F1-F3 and P1-P4. Despite discrepancies in the calculated conductivities of F1-F3 compared with experimental results, the predicted viscosities generally align well with experiments, with the exception of F3. Nonetheless, BAMBOO effectively captures the increasing viscosity trend as the salt concentration rises. Although experimental records of viscosity for P1-P4 are limited, the available results indicate that BAMBOO successfully reproduces experimental viscosities for P1 and P4, consistent with its conductivity performance.

We extract self-diffusivities ($D_{\pm,\text{self}}$) of Li^+ and anions from the simulated MD trajectories and subsequently compute the cation's transference number. The results are tabulated in Table [Supplementary 16](#). The transference number is defined as the ratio of $D_{+,\text{self}}$ over the summation of $D_{+,\text{self}}$ and $D_{-,\text{self}}$ as shown in equation [S65](#). As in Figure [Supplementary 12f](#), our calculation shows that the t_{Li^+} for LiPF_6 -based electrolytes are consistent around 0.4. This finding is in line with existing literature, which reports t_{Li^+} between 0.3 and 0.4 when measured using Bruce-Vincent or pfg-NMR techniques for carbonate-based electrolytes [66]. For LiFSI-based electrolytes, we observed that the t_{Li^+} 's are slightly higher than those in LiPF_6 -based electrolytes, consistent with prior research comparing LiPF_6 and LiFSI in carbonate solvents [67, 68]. Furthermore, we note an increase in t_{Li^+} for LiFSI-based electrolytes as the concentration rises from 1.12m to 3.74m. This observation aligns with the increasing formation of CIPs and AGGs, as illustrated in Figure 3 in the main text. As the fraction of paired ions increases, the self-diffusivities of cations and anions become more correlated and similar, leading the calculated t_{Li^+} to approach 0.5.

8 Supplementary Note: Results of More Practical Liquid Electrolytes

To demonstrate the ability of BAMBOO on simulating practical liquid electrolytes with multiple components, in this section we use the BAMBOO model after density alignment to simulate a four-component system with EC, DMC, PC, and LiPF_6 salt from Zhu *et al.* [69]. In Table [Supplementary 9](#), we list the experimental conductivities from Zhu *et al.* [69] and predicted conductivities from BAMBOO and OPLS-AA. We can see that, for the four-component system, BAMBOO achieves a 25.8% deviation compared with experiment, which is similar to the deviation of 26% for simpler systems listed in Figure [Supplementary 10c](#) and lower than the 47.7% deviation of OPLS-AA.

To further investigate the ability of BAMBOO on predicting properties of high-entropy liquid electrolytes, we conduct experimental measurements on some high-entropy electrolytes with 7 or 8 components as in Table [Supplementary 10](#), and use BAMBOO and OPLS-AA to predict their properties. Here, DEC, EA, EC, EMC are purchased from SHINGHWA ADVANCED MATERIALS GROUP, DMC from Hi-Tech Spring, VC from JIANGSU HUAYI MACHINERY, LiPF_6 from HUNAN ZHONGLAN, and LiFSI from Do-Fluoride New

Materials. All the chemicals are used as received. The electrolytes are prepared at the test environment with the cleanliness class 10,000 at the temperature 24.7 °C. Density is measured by Anton Paar DMA 35 Density Meter, and viscosity is measured by Anton Paar ViscoQC-100L Viscometer.

In Table [Supplementary 10](#), we show the experimentally measured and BAMBOO and OPLS-AA simulated density and viscosity of five high-entropy electrolytes composed of DEC, DMC, EA, EC, EMC, and VC as solvents and LiPF₆ and/or LiFSI as salts. We can see that, the density MAE of BAMBOO is 0.019 g/cm³, which is higher than that (0.011 g/cm³) in Figure [Supplementary 10a](#) for simple solvents and electrolytes. The higher density error of high-entropy electrolytes is expected as the density alignment is performed only on pure solvents and simple electrolytes with one or two solvents and one salt. Despite the increase, the density error of BAMBOO (0.019 g/cm³) for high-entropy electrolytes is still much lower than that of OPLS-AA (0.043 g/cm³). As for viscosity of the high-entropy electrolytes, BAMBOO achieves the mean deviation of 10.1%, which is lower than that (17%) in Figure [Supplementary 10b](#) for simple solvents and electrolytes and that of OPLS-AA (29.4%) for the same high-entropy electrolytes. The lower density and viscosity error of BAMBOO over OPLS-AA shows the potential of BAMBOO for replacing classic force fields with MLFFs to study high-entropy electrolytes.

To further improve the performance of BAMBOO on predicting properties of high-entropy electrolytes, we propose to add their densities into the density alignment algorithm and sample more relevant atomistic structures with DFT labels to supplement the DFT dataset. Due to the exponential complexity of high-entropy electrolytes, when preparing relevant density and DFT data, sampling strategies with higher efficiency should be considered. For example, for density, one might use a surrogate model to predict the output of MD simulations by BAMBOO and evaluate prediction uncertainty, and conduct experimental measurements on the most uncertain compositions. For sampling more DFT data, compared with sampling solvation structures from MD trajectories of electrolytes as in this work, manually constructing dimer structures might be more efficient to ensure that DFT labels of all the pairs of components are included in the DFT dataset.

Table Supplementary 9: The simulated conductivities (σ) of the four-component system with EC, DMC, PC, and LiPF₆ salt from Zhu *et al.* [69] by the BAMBOO MLFF and the OPLS-AA force field [52]. The electrolytes are presented in “solvents|solvent weight %|salt|salt molality” format.

Compositions	σ_{BAMBOO}	$\sigma_{\text{OPLS-AA}}$	σ_{Expt}
DMC_EC_PC 9.9_82 LiPF ₆ 0.87	4.04±0.08	4.87±0.15	7.63±0.02
DMC_EC_PC 70.15_15 LiPF ₆ 0.96	10.87±0.21	6.64±0.97	13.41±0.08
DMC_EC_PC 39.29_33 LiPF ₆ 0.94	6.47±0.12	6.24±0.66	11.52±0.05
DMC_EC_PC 75.18_7 LiPF ₆ 0.96	11.67±0.15	7.02±0.21	13.62±0.01
DMC_EC_PC 70.8_22 LiPF ₆ 0.95	10.74±0.26	6.26±0.23	13.4±0.01
DMC_EC_PC 70.14_16 LiPF ₆ 0.95	11.60±0.32	6.48±0.12	13.41±0.08
DMC_EC_PC 71.21_8 LiPF ₆ 0.97	11.64±0.54	7.45±0.49	13.69±0.08
DMC_EC_PC 40.49_11 LiPF ₆ 0.96	7.46±0.12	6.28±0.43	11.62±0.08
DMC_EC_PC 67.3_29 LiPF ₆ 0.92	9.99±0.39	6.45±0.62	12.65±0.05
DMC_EC_PC 70.9_21 LiPF ₆ 0.95	10.06±0.25	6.66±0.31	13.40±0.01

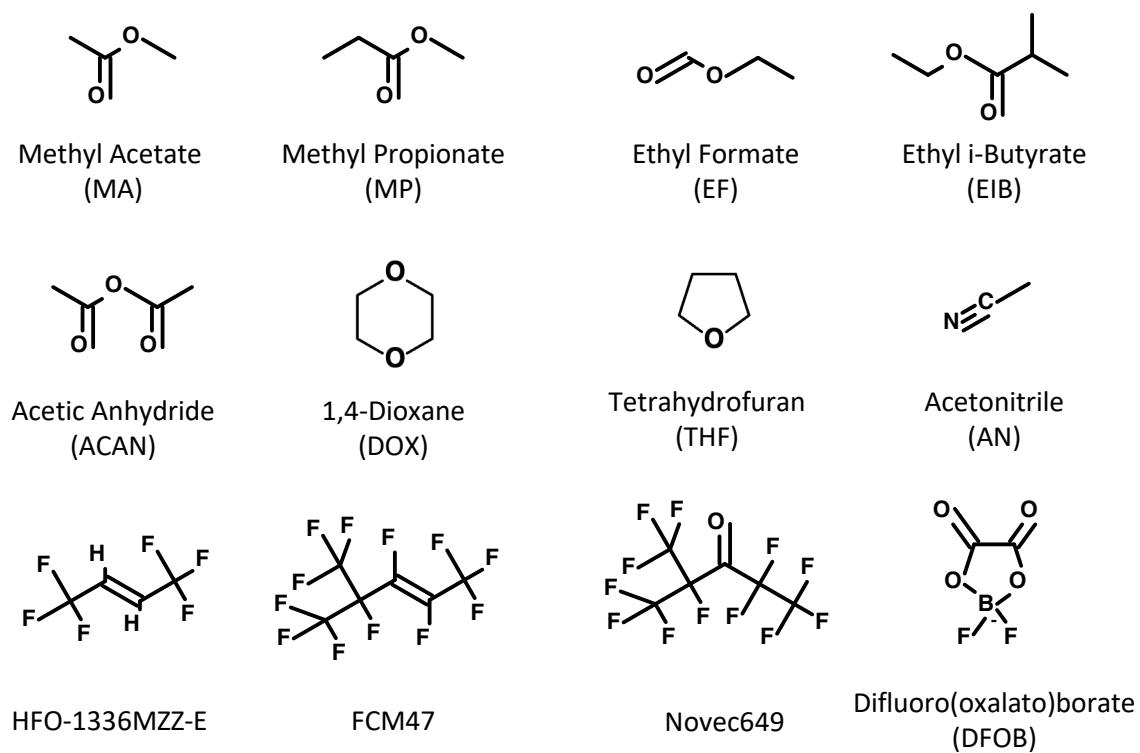


Figure Supplementary 13: Schematic of the unseen species used for testing the transferability of BAMBOO.

Table Supplementary 10: The simulated density (ρ , in g/cm³) and viscosity (η , in cP) of the high-entropy liquid electrolytes. In this table, the compositions are described by eight-component tuples, with the first to the sixth elements denote the weight percentage of DEC, DMC, EA, EC, EMC, and VC, respectively, and the last two elements molality of LiPF₆ and LiFSI, respectively. The uncertainty for the density calculated using BAMBOO and OPLS-AA ranges from 0.001 to 0.001 g/cm³. The uncertainty for the viscosity from BAMBOO ranges from 0.018 to 0.244 cP, while that from OPLS-AA ranges from 0.076 to 0.463 cP.

Compositions	ρ_{BAMBOO}	$\rho_{\text{OPLS-AA}}$	ρ_{Expt}	η_{BAMBOO}	$\eta_{\text{OPLS-AA}}$	η_{Expt}
(8, 6, 6, 67, 7, 6, 0.69, 0)	1.290	1.268	1.317	3.746	5.036	4.409
(8, 6, 12, 61, 7, 6, 0, 0.68)	1.283	1.246	1.295	3.707	4.684	3.931
(8, 6, 6, 67, 7, 6, 0.68, 0.68)	1.349	1.327	1.364	8.467	12.771	9.764
(16, 12, 12, 35, 14, 11, 0.67, 0)	1.207	1.194	1.234	3.046	4.828	3.390
(16, 12, 12, 35, 14, 11, 0, 0.67)	1.220	1.198	1.235	3.223	4.264	3.035

Table Supplementary 11: The stability, density (ρ , in g/cm³), and viscosity (η , in cP) of new unseen molecules/salt in Figure [Supplementary 13](#) from MD simulations powered by BAMBOO and OPLS-AA. The uncertainty for the density calculated using BAMBOO and OPLS-AA ranges from 0.001 to 0.004 g/cm³. The uncertainty for the viscosities calculated using BAMBOO and OPLS-AA ranges from 0.008 to 0.176 cP. For clarity, we do not show simulated properties which cannot be stably simulated by BAMBOO or do not have corresponding experimental values to compare.

Molecule	Stable	ρ_{BAMBOO}	$\rho_{\text{OPLS-AA}}$	ρ_{Exp}	η_{BAMBOO}	$\eta_{\text{OPLS-AA}}$	σ_{Exp}
MA	Yes	1.02	0.953	0.938[70]	0.831	0.610	0.385[70]
MP	Yes	0.984	0.922	0.915[71]	NA	NA	NA
EF	Yes	0.924	0.946	0.922,0.917[72]	0.649	0.637	0.49[73]
EIB	Yes	0.829	0.890	0.865[74]	0.524	0.839	0.56[73]
ACAN	Yes	1.014	1.077	1.080[74]	0.880	1.777	0.85[75]
DOX	Yes	1.021	0.981	1.030[74]	0.673	0.667	1.20[76]
THF	No	NA	NA	NA	NA	NA	NA
AN	No	NA	NA	NA	NA	NA	NA
HFO-1336MZZ-E	Yes	1.488	1.346	1.413[74]	NA	NA	NA
FCM47	Yes	1.571	1.674	1.601[77]	NA	NA	NA
Novec649	Yes	1.670	1.746	1.600[78]	1.381	0.666	0.64[78]
EC LiDFOB 0.25	No	NA	NA	NA	NA	NA	NA

9 Supplementary Note: Transferability to Unseen Molecules

To satisfy the potential interest in exploring new solvent derivatives completely *in silico*, in this section we assess the generalizability of the current BAMBOO model to unseen molecules not included in the DFT dataset, probing its capability limitations and providing guidance for further improvement.

We first conduct MD simulations on 12 new unseen molecules/salt as in Figure [Supplementary 13](#). In Table [Supplementary 11](#), we list whether the BAMBOO-based MD simulations for each species is stable without spurious reactions, and the simulated density and viscosity from BAMBOO and OPLS-AA [52]. We can see that, BAMBOO can stably simulate unseen molecules with bonding types similar to those included in the training set in Figure [Supplementary 1](#). However, we find that for species with unseen element (such as B in DFOB) and unseen bond types (such as the triple bond in AN and the dihedral of four carbon atoms in THF), BAMBOO cannot achieve long-time stable MD simulations. For the remaining molecules that can be stably simulated by BAMBOO, we can see that BAMBOO has the density MAE of 0.0495 g/cm³ and the mean

viscosity deviation of 52.4%, which is higher than that of systems included in the training set (0.011 g/cm^3 and 16.7%). As comparison, for the same systems in Table [Supplementary 11](#), the density MAE of OPLS-AA is 0.0454 g/cm^3 and the mean viscosity deviation of OPLS-AA is 49.3%. In practice, we recommend that for unseen species, people should first use classic force fields such as OPLS-AA to obtain rough estimation of bulk properties as OPLS-AA restricts bond-breaking from happening, then construct DFT datasets and train machine learning force fields such as BAMBOO to more accurately study molecules of interest.

In Figure [Supplementary 11](#), we show the strong linear dependence of viscosity and ionic conductivity on density: a denser liquid exhibits stronger inter-molecular interactions, resulting in higher viscosity and lower conductivity. From Table [Supplementary 11](#), we observe that, for systems where BAMBOO considerably overestimates densities compared with OPLS-AA and experiment, such as MA and Novec649, we also observe overestimated viscosities from BAMBOO. The simultaneous overestimation of density and viscosity aligns with the relation between density and viscosity, which also implies that, for unseen molecules, the tendency of BAMBOO to overestimate or underestimate viscosity could be speculated by the behavior of density prediction. As for ionic conductivity, we can also speculate its transferability from the behaviors of density and viscosity predictions. For example, since BAMBOO overestimates the density and viscosity of MA, we expect that the ionic conductivity of electrolytes containing MA would be underestimated. For example, for the system DMC_EC_MA|30-40-30|LiPF₆|0.50, its experimental viscosity and conductivity are 1.09 cP and 12.83 mS/cm, respectively [63]. As anticipated, BAMBOO overestimates its viscosity ($1.80 \pm 0.02\text{ cP}$) and underestimates its conductivity ($10.45 \pm 0.35\text{ mS/cm}$).

To further reveal the trend of density prediction on unseen molecules, we further conduct MD simulations on fluorinated solvent molecules. In advanced electrolytes, fluorination of a base solvent is a commonly used strategy [79, 80] to enhance electrochemical performance [81], LiF-rich SEI formation [82], and fire-resisting property [83]. Despite the popularity and effectiveness of fluorination, the exact relationship of the location and the degree of fluorination to the liquid electrolyte performances are not clear. This underscores the importance to incorporate MLFF in structural-based electrolyte design. The parent molecules, EC, DMC, and EA and their derived child molecular structures are shown in Figure [Supplementary 14a, c, and e](#). All child molecules are excluded from training or alignment, except FEC. The same BAMBOO model used for generating the liquid electrolyte properties shown in Figure [Supplementary 12](#) is used in the transferability study. We find all simulations of the child solvents remain stable during the 5 ns MD simulations within a broad temperature range of 283-343 K, demonstrating the high transferability of the model.

In Figure [Supplementary 14b, d, and f](#), we present the simulated densities of the fluorinated derivatives from BAMBOO MD simulations and experiments as a function of temperatures. Qualitatively, we find that, except for the fluorinated EAs (FEA, FEA), which have very similar structures and experimental densities, BAMBOO can correctly predict the density trends of fluorinated ECs and DMCs. To be specific, the trend cis-DFEC > DFEC > trans-DEFC, and the trend FDMC > DFDMC > MFDMC are reproduced. The trend for DMC derivatives follows the intuition that as the fluorination increases, density increases. In addition, we find that the more structural-alike the unseen child molecule compared with the parent molecule, the more

accurate the prediction of density from BAMBOO is. Furthermore, the inclusion of the functional group of interest in training data is crucial for accuracy. For instance, since FEC is already included in the DFT dataset, the density discrepancy of other fluorinated derivatives of EC from BAMBOO are within 0.03 g/cm^3 compared with that from experiments. As a comparison, since there is no fluorinated derivative of EA and DMC in the DFT dataset, the density errors of the simulated results for fluorinated EA and DMC are generally at the order of 0.1 g/cm^3 , which is larger than that of fluorinated EC. Specifically, with higher degree of fluorination, we can see that the density error of DMC's child molecules becomes larger, from around 0.05 g/cm^3 for MFDMC, 0.08 g/cm^3 for DFDMC, to 0.20 g/cm^3 for TFDMC. The comparison between EC and DMC derived molecules indicate that BAMBOO has the potential to transfer to unseen molecules in terms of calculating density but requires small amount of DFT training data describing the specific functional groups and types of molecular backbone. Here, we speculate the accuracy of density predictions to the fluorinated EAs and DMCs can be improved by including “-CH₂F” and “-CHF₂” functional groups in the DFT dataset, which will be incorporated in a later version of BAMBOO.

To make machine learning force field more transferable to out-of-sample systems, we need constant effort to improve the MLFFs. In general, there are two approaches to elevate the transferability of MLFFs. The first approach is to conduct large scale pretraining on datasets with diverse molecules, such as the MACE-OFF23 [84] which is trained on nearly a million conformers. Although the strategy of large scale pretraining is very popular in the recent wave of universal force fields [35, 85, 86, 87, 88, 89, 84], it requires careful design of the dataset for the trade-off between effectiveness and efficiency, and it is not proved to guarantee stable MD simulations for all molecules. The other approach is to combine machine learning model with strong physical restrictions, such as our latest work that employs machine learning models to parameterize force fields with classic functional forms [90], which ensures that the force field can stably simulate any molecule without spurious reaction during the MD simulation. However, such restrictions also limit the usage of machine learning force field on cases where bond breaking happens. Therefore, we think it is important to further develop machine learning force field that can stably simulate as many molecules as possible while preserve the ability to model bond breaking events when necessary. In the very recent work by Niblett *et al.* [91], it is shown that the GNN based model is more transferable to new molecules than local descriptors-based model, which we hope can inspire further development of machine learning architecture for higher transferability of force fields.

10 Supplementary Note: More Solvation Structure Analyses

The charge histograms shown in Figure 3 in the main text and Figure [Supplementary 15](#) are derived by first collecting atomic partial charges from each frame of a 4ns simulation, totalling 4,000 frames, followed by histogram analyses. The peaks corresponding to each normal distribution in the figure panels are identified using the *find-peaks* function as implemented in *scipy.signal* module. The solvation structures were determined by extracting Li clusters using *MDAnalysis* [92] with radius cut-off 2.2Å determined from Li-FSI radial distribution functions. This information can be read from the cumulative coordination around Li⁺ presented in Figure

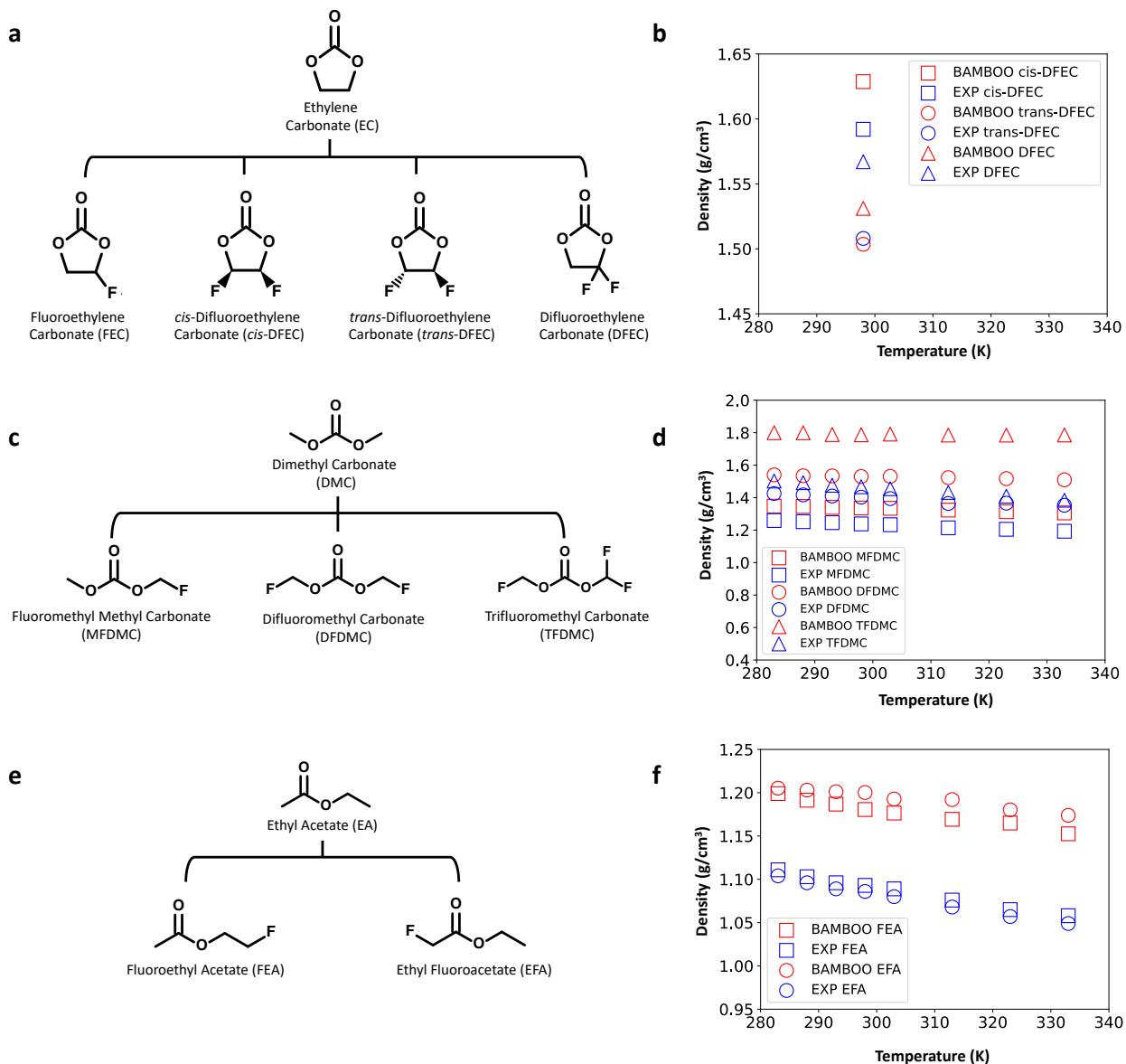


Figure Supplementary 14: Transferability of BAMBOO to unseen fluorinated molecules. **a, b.** Illustration of fluorinated EC molecules and densities of the EC derivatives from BAMBOO and experiment, respectively. **c, d.** Illustration of fluorinated DMC molecules and densities of the DMC derivatives from BAMBOO and experiment, respectively. **e, f.** Illustration of fluorinated EA molecules and densities of the EA derivatives from BAMBOO and experiment, respectively.

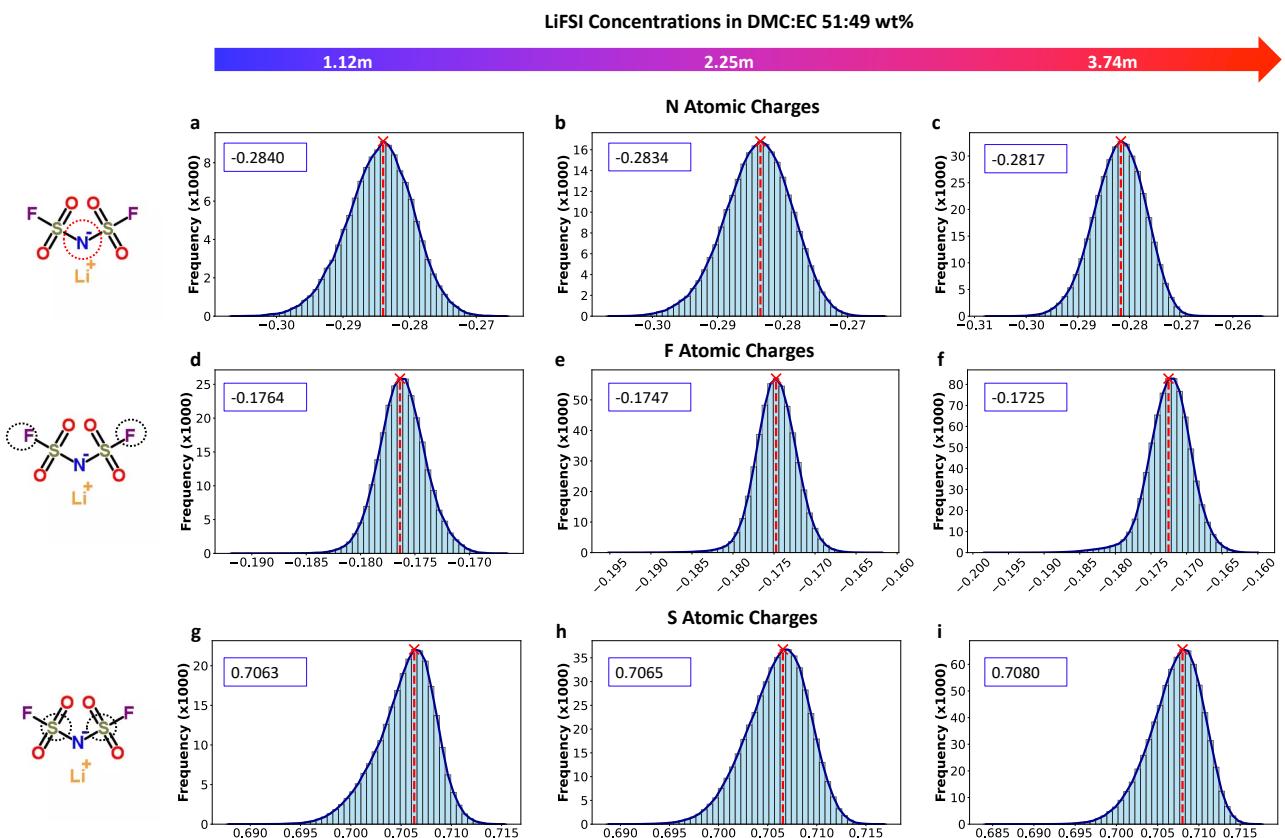


Figure Supplementary 15: The atomic charge distributions of N, S, and F of the three simulated LiFSI electrolytes using BAMBOO MLFF. The electrolytes analyzed are the same systems as those shown in Figure 3 in the main text and F1-F3 electrolytes shown in Figure Supplementary 12.

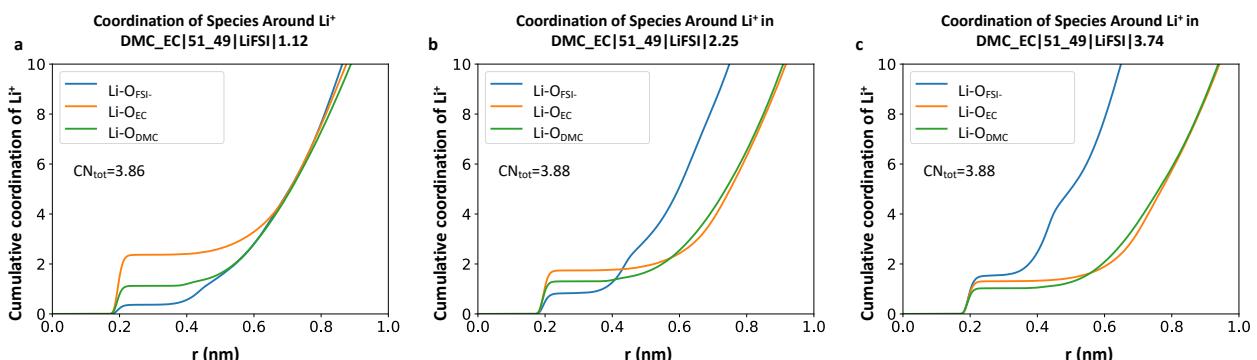


Figure Supplementary 16: The cumulative coordination of solvent and anion species around Li⁺. **a-c** show the results of electrolytes 1.12m, 2.25m, and 3.74m LiFSI in DMC:EC 51:49 wt%, respectively. The total coordination numbers for each simulated system is presented in each figure panel. The legends indicate the coordination is obtained by analyzing the atomise interactions between Li⁺ and the Os of surrounding solvents and anion molecules. The electrolytes analyzed are the same systems as those shown in Figure 3 in the main text and F1-F3 electrolytes shown in Figure Supplementary 12.

Supplementary 16. The frames involved in the cluster extraction are in consistent with the charge histograms.

In Figure 3 in the main text, we observe that with increasing concentration, there is a rise in the fraction of Li ions with lower atomic charges. O_{FSI}⁻ also shows such a trend. This observation may initially appear counter-intuitive, as it suggests the Li charge decreases did not end up being taken by O_{FSI}⁻s. Upon analyzing the charge histograms of N, F, and S in Figure Supplementary 15, we observe consistent shifts in the distribution peaks from more negative to more positive values. These shifts, totaling 0.014, likely indicate that atomic

charges moved away from Li ions being compensated by N, F, and S, along with other solvent molecules.

The results presented in this work show that the charges of Li^+ and other atomic species are not fixed, rather, they change as electrolyte compositions. In liquid electrolyte simulations, accurately capturing polarizability is crucial for describing Li^+ transport properties. However, classical force fields often fall short in this aspect. A typical treatment in classical MD using a non-polarizable force field is to scale the ions' and solvents' charges [93, 94] to match experimental properties, such as self-diffusivities for specific systems of interest. The goal is to provide a mean-field representation of charge screening [95]. However, different scaling factors are required when investigating different electrolyte chemistries or compositions, leading to limited generalizability. Determining the optimal charge scaling scheme typically relies on Li^+ self-diffusivity data obtained from Nuclear Magnetic Resonance Spectroscopy (NMR) [94], which can be costly and time-consuming, and sometimes defeat the purpose of new liquid electrolyte exploration. Therefore, given the generalizability and flexibility of BAMBOO in providing solvation insights, there exists an unique opportunity for MLFF to participate in “bottom-up” understanding [95] and design of liquid electrolytes.

11 Supplementary Note: Liquid Electrolyte Data

In this section, we show all the simulated and experimental properties of solvents and solutions for readers’ reference.

Table Supplementary 12: The simulated densities in g/cm^3 of pure solvents and mixtures that are used in the density alignment process. The salt concentrations (“m”) of mixtures are in molality, defined as the mole of salt in kilograms of solvent. The compositions of the LiFSI-containing mixtures are converted based on their molar ratios as used in the original literature. The electrolytes are presented in “solvents|solvent weight %|salt|salt molality” format. The uncertainty associated with the density prediction ranges from 0.001-0.003 g/cm^3 . All simulations are performed at 298K, except for EC and FEC, which were conducted at 313K.

Mixtures	Compositions	BAMBOO	Exp	Exp. Ref.
DMC	DMC 100 wt%	1.060	1.06	[58]
DEC	DEC 100 wt%	0.969	0.97	[58]
PC	PC 100 wt%	1.188	1.20, 1.194	[96, 51]
EC	EC 100 wt%	1.310	1.32	[58]
FEC	FEC 100 wt%	1.474	1.48	[60]
EA	EA 100 wt%	0.892	0.90	[97]
Novec7000	Novec7000 100 wt%	1.416	1.41	[98]
P5	DMC_EC 60_40 LiPF ₆ 0.9	1.236	1.239	Acquired from [99]
P6	DMC_EC_EMG 45_50_5 LiPF ₆ 1.1	1.274	1.273	Acquired from [99]
P7	DMC_EC 70_30 LiPF ₆ 1.5	1.255	1.268	Acquired from [99]
F4	DMC LiFSI 2.22	1.260	1.27	DMC:LiFSI 5:1 [100]
F5	EC LiFSI 0.49	1.359	1.38	EC:LiFSI 23:1 [100]
F6	EC LiFSI 1.14	1.412	1.43	EC:LiFSI 10:1 [100]

References

- [1] Adam Paszke et al. “Pytorch: An imperative style, high-performance deep learning library”. In: *Advances in neural information processing systems* 32 (2019).

Table Supplementary 13: The simulated densities in g/cm³ of pure solvents and mixtures that are **not** used in the density alignment process. The salt concentrations (“m”) of mixtures are in molality, defined as the mole of salt in kilograms of solvent. The compositions of the LiFSI-containing mixtures are converted based on their molar ratios as used in the original literature. The electrolytes are presented in “solvents|solvent weight %|salt|salt molality” format. The uncertainty associated with the density prediction ranges from 0.001-0.003 g/cm³. The simulations are conducted at 298K.

Mixtures	Compositions	BAMBOO	Exp	Exp. Ref.
EMC	EMC 100 wt%	1.015	1.0	[58]
VC	VC 100 wt %	1.370	1.355	[58]
EO	EO 100 wt %	0.769	0.79	[74]
ACT	ACT 100 wt %	0.808	0.79	[74]
EP	EP 100 wt %	0.919	0.884	[101]
S1	DMC_EC 70_30	1.131	1.13	SI of [99]
S2	DMC_EC 50_50	1.183	1.18	SI of [99]
S3	EMC_EC 70_30	1.100	1.095	SI of [99]
S4	EMC_EC 50_50	1.160	1.156	SI of [99]
P1	DMC_EC 50_50 LiPF ₆ 0.5	1.224	1.235	Acquired from [99]
P4	DMC_EC 70_30 LiPF ₆ 1.0	1.221	1.228	Acquired from [99]
P8	DMC_EC 70_30 LiPF ₆ 1.3	1.243	1.252	Acquired from [99]
F7	DMC LiFSI 3.70	1.361	1.36	DMC:LiFSI 3:1 [100]
F8	DMC LiFSI 1.11	1.167	1.18	DMC:LiFSI 10:1 [100]
F9	EC LiFSI 3.78	1.555	1.57	EC:LiFSI 3:1 [100]
F10	EC LiFSI 2.27	1.484	1.50	EC:LiFSI 5:1 [100]
F3	DMC_EC 51_49 LiFSI 3.74	1.456	1.46	DMC:EC:LiFSI 3:3:2 [100]
F2	DMC_EC 51_49 LiFSI 2.25	1.372	1.38	DMC:EC:LiFSI 5:5:2 [100]
F1	DMC_EC 51_49 LiFSI 1.12	1.285	1.30	DMC:EC:LiFSI 10:10:2 [100]
F11	DMC_EC 51_49 LiFSI 0.49	1.229	1.24	DMC:EC:LiFSI 23:23:2 [100]

Table Supplementary 14: The simulated viscosities in cP of solvents

Mixtures	Compositions	Temperature, K	BAMBOO	Exp
DMC	DMC 100 wt%	298	0.629±0.0290	0.585,0.706[58, 59]
DEC	DEC 100 wt%	298	0.766±0.0243	0.75[58]
EMC	EMC 100 wt%	298	0.724±0.0204	0.65[58]
VC	VC 100 wt%	298	1.104±0.0540	1.78[61]
EC	EC 100 wt%	313	1.381±0.0386	1.60-1.90[58, 60, 59]
FEC	FEC 100 wt%	313	2.349±0.0941	2.24-4.1[60, 62, 59]
Novec7000	Novec7000 100 wt%	298	0.447±0.0211	0.45[98]
EA	EA 100 wt%	298	0.542±0.0071	0.43[102]
PC	PC 100 wt%	298	1.426±0.0452	2.53[58]
EO	EO 100 wt%	298	0.829±0.0500	1.1[74]
ACT	ACT 100 wt%	298	0.502±0.0101	0.31[74]
EP	EP 100 wt%	298	0.728±0.040	0.63[101]
S1	DMC_EC 70_30	293	0.720±0.0136	0.95[63]
S1	DMC_EC 70_30	300	0.673±0.0171	NA
S1	DMC_EC 70_30	303	0.644±0.0197	0.83[63]
S1	DMC_EC 70_30	313	0.602±0.0129	0.73[63]
S3	EMC_EC 70_30	293	0.842±0.0126	1.11[63]
S3	EMC_EC 70_30	300	0.820±0.0163	NA
S3	EMC_EC 70_30	303	0.835±0.0253	0.97[63]
S3	EMC_EC 70_30	313	0.734±0.0161	0.87[63]

Table Supplementary 15: The simulated viscosities (η , in cP), ionic conductivities (σ , in mS/cm) of cation of LiFSI and LiPF₆ based electrolytes at 298K. The uncertainty for the conductivities calculated using Mistry's method ranges from 0.35 to 1.38 mS/cm, while the values calculated using NE method ranges from 0.11 to 2.20 mS/cm. The uncertainty for the predicted viscosity for medium and low concentrated electrolytes (P1-P9, and F1, F5-6, F8, F11) are below 0.1 cP, while the uncertainty for the remaining highly concentrated electrolytes are generally from 0.3 cP to 0.6 cP with the maximum of 3 cP for F9, respectively. The viscosity values for the carbonate based electrolytes are acquired from authors of ref[99] by private communications.

Mixtures	Compositions	η_{simu}	η_{Exp}	σ_{Mistry}	σ_{NE}	σ_{Exp}
P1	DMC_EC 50_50 LiPF ₆ 0.5	1.891	2.219[99]	11.201	12.420	12.14[99]
P2	DMC_EC 70_30 LiPF ₆ 0.5	1.656	NA	11.361	13.133	10.89[99]
P3	DMC_EC 70_30 LiPF ₆ 0.84	2.413	NA	11.990	15.106	12.91[99]
P4	DMC_EC 70_30 LiPF ₆ 1.0	2.683	2.734[99]	12.881	15.451	12.43[99]
P5	DMC_EC 60_40 LiPF ₆ 0.9	2.847	2.778[99]	11.531	14.860	12.793[99]
P6	DMC_EC_EMC 45_50_5 LiPF ₆ 1.1	4.716	4.566[99]	8.021	10.669	12.175[99]
P7	DMC_EC 70_30 LiPF ₆ 1.5	4.810	4.472[99]	8.713	12.241	12.144[99]
P8	DMC_EC 70_30 LiPF ₆ 1.3	3.984	3.767[99]	9.573	13.171	12.118[99]
P9	PC LiPF ₆ 0.28	2.517	NA	4.735	5.053	4.967[103]
F1	DMC_EC 51_49 LiFSI 1.12	3.548	3.4 [100]	9.828	13.069	14[100]
F2	DMC_EC 51_49 LiFSI 2.25	11.121	9.8 [100]	4.369	6.592	9.8[100]
F3	DMC_EC 51_49 LiFSI 3.74	24.946	36.9 [100]	2.266	3.527	4.5[100]
F4	DMC LiFSI 2.22	4.653	3.9 [100]	9.237	13.142	12.2[100]
F5	EC LiFSI 0.49	3.184	4.1 [100]	7.013	8.099	8.7[100]
F6	EC LiFSI 1.14	6.179	8.1 [100]	5.907	7.672	9.7[100]
F7	DMC LiFSI 3.70	13.304	12.9 [100]	4.363	6.739	8.1[100]
F8	DMC LiFSI 1.11	1.629	1.5 [100]	15.984	20.209	9.9[100]
F9	EC LiFSI 3.78	34.183	NA	1.738	2.472	2.3 [100]
F10	EC LiFSI 2.27	17.283	33.1 [100]	2.597	3.063	5.6[100]
F11	DMC_EC 51_49 LiFSI 0.49	1.85	1.9 [100]	11.015	12.771	11.5[100]

Table Supplementary 16: The simulated diffusion coefficients (10^{-10} m²/s) of cations and anions, and the calculated transference numbers in LiPF₆ and LiFSI-based electrolytes at 298K.

Mixtures	Compositions	D _{+,self}	D _{-,self}	t ₊
P1	DMC_EC 50_50 LiPF ₆ 0.5	2.898	4.298	0.403±0.031
P2	DMC_EC 70_30 LiPF ₆ 0.5	3.256	4.651	0.412±0.028
P3	DMC_EC 70_30 LiPF ₆ 0.84	2.587	3.493	0.425±0.027
P4	DMC_EC 70_30 LiPF ₆ 1.0	2.435	3.307	0.424±0.035
F1	DMC_EC 51_49 LiFSI 1.12	1.919	2.412	0.443±0.046
F2	DMC_EC 51_49 LiFSI 2.25	0.875	0.952	0.479±0.055
F3	DMC_EC 51_49 LiFSI 3.74	0.548	0.550	0.499±0.056

- [2] Oliver Unke and Markus Meuwly. “PhysNet: A Neural Network for Predicting Energies, Forces, Dipole Moments and Partial Charges”. In: *J. Chem. Theory Comput.* 15 (May 2019). DOI: [10.1021/acs.jctc.9b00181](https://doi.org/10.1021/acs.jctc.9b00181).
- [3] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. “Sigmoid-weighted linear units for neural network function approximation in reinforcement learning”. In: *Neural networks* 107 (2018), pp. 3–11.
- [4] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. “Layer normalization”. In: *arXiv preprint arXiv:1607.06450* (2016).

- [5] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017).
- [6] Ilyes Batatia et al. *The Design Space of E(3)-Equivariant Atom-Centered Interatomic Potentials*. 2022. arXiv: [2205.06643 \[stat.ML\]](https://arxiv.org/abs/2205.06643).
- [7] Keqiang Yan et al. *Periodic Graph Transformers for Crystal Material Property Prediction*. 2022. arXiv: [2209.11807 \[cs.LG\]](https://arxiv.org/abs/2209.11807).
- [8] Pier Poier et al. “Molecular Dynamics Using Nonvariational Polarizable Force Fields: Theory, Periodic Boundary Conditions Implementation, and Application to the Bond Capacity Model”. In: *J. Chem. Theory Comput.* 2019 (Sept. 2019). doi: [10.1021/acs.jctc.9b00721](https://doi.org/10.1021/acs.jctc.9b00721).
- [9] Heiner Schröder, Anne Creon, and Tobias Schwabe. “Reformulation of the D3(Becke–Johnson) Dispersion Correction without Resorting to Higher than C6 Dispersion Coefficients”. In: *J. Chem. Theory Comput.* 11.7 (June 2015), pp. 3163–3170. doi: [10.1021/acs.jctc.5b00400](https://doi.org/10.1021/acs.jctc.5b00400).
- [10] Alejandro Torres-Sánchez, Juan M. Vanegas, and Marino Arroyo. “Geometric derivation of the microscopic stress: A covariant central force decomposition”. In: *J. Mech. Phys. Solids* 93 (2016). Special Issue in honor of Michael Ortiz, pp. 224–239. ISSN: 0022-5096. doi: <https://doi.org/10.1016/j.jmps.2016.03.006>.
- [11] Zhifeng Jing et al. “Polarizable Force Fields for Biomolecular Simulations: Recent Advances and Applications”. In: *Annual Review of Biophysics* 48. Volume 48, 2019 (2019), pp. 371–394. ISSN: 1936-1238. doi: <https://doi.org/10.1146/annurev-biophys-070317-033349>.
- [12] Dmitry Bedrov et al. “Molecular Dynamics Simulations of Ionic Liquids and Electrolytes Using Polarizable Force Fields”. In: *Chemical Reviews* 119.13 (2019). PMID: 31141351, pp. 7940–7995. doi: [10.1021/acs.chemrev.8b00763](https://doi.org/10.1021/acs.chemrev.8b00763). URL: <https://doi.org/10.1021/acs.chemrev.8b00763>.
- [13] A. P. Thompson et al. “LAMMPS – a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales”. In: *Comp. Phys. Comm.* 271 (2022), p. 108171. doi: [10.1016/j.cpc.2021.108171](https://doi.org/10.1016/j.cpc.2021.108171).
- [14] Brad A. Wells and Alan L. Chaffee. “Ewald Summation for Molecular Simulations”. In: *J. Chem. Theory Comput.* 11.8 (2015). PMID: 26574452, pp. 3684–3695. doi: [10.1021/acs.jctc.5b00093](https://doi.org/10.1021/acs.jctc.5b00093).
- [15] Taiping Hu et al. *Direct Observation of Dendrites Nucleation in Li Metal Battery by Machine Learning Accelerated Molecular Simulations under Realistic Electrochemical Conditions*. 2024. arXiv: [2406.14025 \[cond-mat.mtrl-sci\]](https://arxiv.org/abs/2406.14025). URL: <https://arxiv.org/abs/2406.14025>.
- [16] Leif Jacobson et al. “Transferable Neural Network Potential Energy Surfaces for Closed-Shell Organic Molecules: Extension to Ions”. In: *J. Chem. Theory Comput.* 18 (Mar. 2022). doi: [10.1021/acs.jctc.1c00821](https://doi.org/10.1021/acs.jctc.1c00821).

- [17] Curt M. Breneman and Kenneth B. Wiberg. “Determining atom-centered monopoles from molecular electrostatic potentials. The need for high sampling density in formamide conformational analysis”. In: *J. Comput. Chem.* 11.3 (1990), pp. 361–373. doi: <https://doi.org/10.1002/jcc.540110311>.
- [18] URL: <https://docs.eyesopen.com/toolkits/python/quacpactk/molchargetheory.html>.
- [19] RS Mulliken. *Electronic population analysis on LCAO-MO molecular wave functions. IJ Chem Phys* 23: 1833–1840. 1955.
- [20] Toon Verstraelen et al. “Minimal basis iterative stockholder: atoms in molecules for force-field development”. In: *Journal of Chemical Theory and Computation* 12.8 (2016), pp. 3894–3912.
- [21] Nicola Marzari et al. “Maximally localized Wannier functions: Theory and applications”. In: *Reviews of Modern Physics* 84.4 (2012), pp. 1419–1475.
- [22] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: [1412.6980 \[cs.LG\]](https://arxiv.org/abs/1412.6980).
- [23] Z. Mu et al. *BAMBOO*. 2024. URL: <https://github.com/bytedance/bamboo>.
- [24] Axel D. Becke. “Density-functional thermochemistry. III. The role of exact exchange”. In: *J. Chem. Phys.* 98.7 (Apr. 1993), pp. 5648–5652. ISSN: 0021-9606. doi: [10.1063/1.464913](https://doi.org/10.1063/1.464913).
- [25] Armin Hellweg and Dmitrij Rappoport. “Development of new auxiliary basis functions of the Karlsruhe segmented contracted basis sets including diffuse basis functions (def2-SVPD, def2-TZVPPD, and def2-QVPPD) for RI-MP2 and RI-CC calculations”. In: *Phys. Chem. Chem. Phys.* 17 (2 2015), pp. 1010–1017. doi: [10.1039/C4CP04286G](https://doi.org/10.1039/C4CP04286G).
- [26] Florian Weigend. “Hartree–Fock exchange fitting basis sets for H to Rn”. In: *J. Comput. Chem.* 29.2 (2008), pp. 167–175.
- [27] X. Wu et al. *GPU4PySCF*. 2023. URL: <https://github.com/pyscf/gpu4pyscf>.
- [28] Xiaojie Wu et al. *Python-Based Quantum Chemistry Calculations with GPU Acceleration*. 2024. arXiv: [2404.09452 \[physics.comp-ph\]](https://arxiv.org/abs/2404.09452).
- [29] Yihan Shao et al. “Advances in molecular quantum chemistry contained in the Q-Chem 4 program package”. In: *Mol. Phys.* 113.2 (2015), pp. 184–215. doi: [10.1080/00268976.2014.952696](https://doi.org/10.1080/00268976.2014.952696).
- [30] Steven Dajnowicz et al. “High-Dimensional Neural Network Potential for Liquid Electrolyte Simulations”. In: *J. Phys. Chem. B* 126 (Aug. 2022). doi: [10.1021/acs.jpcb.2c03746](https://doi.org/10.1021/acs.jpcb.2c03746).
- [31] Xiang Fu et al. “Forces are not enough: Benchmark and critical evaluation for machine learning force fields with molecular simulations”. In: *arXiv preprint arXiv:2210.07237* (2022).
- [32] Georg Martius and Christoph H Lampert. “Extrapolation and learning equations”. In: *arXiv preprint arXiv:1610.02995* (2016).

- [33] Pushun Lu et al. “Amorphous bimetallic polysulfide for all-solid-state batteries with superior capacity and low-temperature tolerance”. In: *Nano Energy* 118 (2023), p. 109029. ISSN: 2211-2855. DOI: <https://doi.org/10.1016/j.nanoen.2023.109029>.
- [34] Anubhav Jain et al. “Commentary: The Materials Project: A materials genome approach to accelerating materials innovation”. In: *APL Mater.* 1.1 (July 2013), p. 011002. ISSN: 2166-532X. DOI: [10.1063/1.4812323](https://doi.org/10.1063/1.4812323).
- [35] Chi Chen and Shyue Ong. “A universal graph deep learning interatomic potential for the periodic table”. In: *Nat. Comput. Sci.* 2 (Nov. 2022), pp. 718–728. DOI: [10.1038/s43588-022-00349-3](https://doi.org/10.1038/s43588-022-00349-3).
- [36] Paolo Giannozzi et al. “QUANTUM ESPRESSO: a modular and open-source software project for quantum simulations of materials”. In: *J. Phys. Condens. Matter* 21.39 (Sept. 2009), p. 395502. DOI: [10.1088/0953-8984/21/39/395502](https://doi.org/10.1088/0953-8984/21/39/395502).
- [37] John P. Perdew, Kieron Burke, and Matthias Ernzerhof. “Generalized Gradient Approximation Made Simple”. In: *Phys. Rev. Lett.* 77 (18 Oct. 1996), pp. 3865–3868. DOI: [10.1103/PhysRevLett.77.3865](https://doi.org/10.1103/PhysRevLett.77.3865).
- [38] L.D. LANDAU and E.M. LIFSHITZ. “CHAPTER XV - SCATTERING OF ELECTROMAGNETIC WAVES”. In: *Electrodynamics of Continuous Media (Second Edition)*. Ed. by L.D. LANDAU and E.M. LIFSHITZ. Second Edition. Vol. 8. Course of Theoretical Physics. Amsterdam: Pergamon, 1984, pp. 413–438. ISBN: 978-0-08-030275-1. DOI: <https://doi.org/10.1016/B978-0-08-030275-1.50021-7>.
- [39] Yu. A. Atanov and A. I. Berdenikov. “Relation between fluid viscosity and compressibility”. In: *J. Eng. Phys.* 43.2 (Aug. 1982), pp. 878–883. DOI: [10.1007/BF00825016](https://doi.org/10.1007/BF00825016).
- [40] “2 - PVT Tests and Correlations”. In: *PVT and Phase Behaviour of Petroleum Reservoir Fluids*. Ed. by Ali Danesh. Vol. 47. Developments in Petroleum Science. Elsevier, 1998, pp. 33–104. DOI: [https://doi.org/10.1016/S0376-7361\(98\)80024-1](https://doi.org/10.1016/S0376-7361(98)80024-1).
- [41] Xinyan Wang et al. “DMFF: An Open-Source Automatic Differentiable Platform for Molecular Force Field Development and Molecular Dynamics Simulation”. In: *J. Chem. Theory Comput.* 19.17 (2023). PMID: 37589304, pp. 5897–5909. DOI: [10.1021/acs.jctc.2c01297](https://doi.org/10.1021/acs.jctc.2c01297).
- [42] Sébastien Röcken and Julija Zavadlav. “Accurate machine learning force fields via experimental and simulation data fusion”. In: *npj Computational Materials* 10.1 (2024), p. 69.
- [43] Davide Tisi et al. “Heat transport in liquid water from first-principles and deep neural network simulations”. In: *Phys. Rev. B* 104 (22 Dec. 2021), p. 224202. DOI: [10.1103/PhysRevB.104.224202](https://doi.org/10.1103/PhysRevB.104.224202). URL: <https://link.aps.org/doi/10.1103/PhysRevB.104.224202>.
- [44] Sheng Gong et al. “Examining graph neural networks for crystal structures: Limitations and opportunities for capturing periodicity”. In: *Sci. Adv.* 9.45 (2023), eadi3245. DOI: [10.1126/sciadv.adl3245](https://doi.org/10.1126/sciadv.adl3245).
- [45] Yi-Lun Liao et al. *EquiformerV2: Improved Equivariant Transformer for Scaling to Higher-Degree Representations*. 2024. arXiv: [2306.12059 \[cs.LG\]](https://arxiv.org/abs/2306.12059). URL: <https://arxiv.org/abs/2306.12059>.

- [46] Philipp Thölke and Gianni De Fabritiis. *TorchMD-NET: Equivariant Transformers for Neural Network based Molecular Potentials*. 2022. arXiv: [2202.02541 \[cs.LG\]](https://arxiv.org/abs/2202.02541).
- [47] Jake Topping et al. *Understanding over-squashing and bottlenecks on graphs via curvature*. 2022. arXiv: [2111.14522 \[stat.ML\]](https://arxiv.org/abs/2111.14522). URL: <https://arxiv.org/abs/2111.14522>.
- [48] Chigozie Nwankpa et al. “Activation functions: Comparison of trends in practice and research for deep learning”. In: *arXiv preprint arXiv:1811.03378* (2018).
- [49] Aashutosh Mistry et al. “On Relative Importance of Vehicular and Structural Motions in Defining Electrolyte Transport”. In: *J. Electrochem. Soc.* 170.11 (2023), p. 110536.
- [50] Edward J. Maginn et al. “Best Practices for Computing Transport Properties 1. Self-Diffusivity and Viscosity from Equilibrium Molecular Dynamics [Article v1.0]”. In: *Living Journal of Computational Molecular Science* 1.1 (Dec. 2018), p. 6324. DOI: [10.33011/livecoms.1.1.6324](https://doi.org/10.33011/livecoms.1.1.6324).
- [51] Oleg Borodin. “Polarizable Force Field Development and Molecular Dynamics Simulations of Ionic Liquids”. In: *J. Phys. Chem. B* 113.33 (2009). PMID: 19637900, pp. 11463–11478. DOI: [10.1021/jp905220k](https://doi.org/10.1021/jp905220k).
- [52] William L. Jorgensen, David S. Maxwell, and Julian Tirado-Rives. “Development and Testing of the OPLS All-Atom Force Field on Conformational Energetics and Properties of Organic Liquids”. In: *Journal of the American Chemical Society* 118.45 (1996), pp. 11225–11236.
- [53] Jean-Christophe Soetens, Claude Millot, and Bernard Maigret. “Molecular dynamics simulation of Li+ BF4-in ethylene carbonate, propylene carbonate, and dimethyl carbonate solvents”. In: *The Journal of Physical Chemistry A* 102.7 (1998), pp. 1055–1061.
- [54] Somisetty V Sambasivarao and Orlando Acevedo. “Development of OPLS-AA force field parameters for 68 unique ionic liquids”. In: *Journal of chemical theory and computation* 5.4 (2009), pp. 1038–1050.
- [55] K Oldiges et al. “Understanding transport mechanisms in ionic liquid/carbonate solvent electrolyte blends”. In: *Physical Chemistry Chemical Physics* 20.24 (2018), pp. 16579–16591.
- [56] *Comparison of Simulation and Experiment*. URL: <https://wasatchmolecular.com/tables.pdf>.
- [57] Christopher L Berhaut et al. “Ionic association analysis of LiTDI, LiFSI and LiPF 6 in EC/DMC for better Li-ion battery performances”. In: *RSC advances* 9.8 (2019), pp. 4599–4608.
- [58] Kang Xu. “Nonaqueous liquid electrolytes for lithium-based rechargeable batteries”. In: *Chem. Rev.* 104.10 (2004), pp. 4303–4418.
- [59] Yukio Sasaki. “Chapter 13 - Physical and electrochemical properties and application to lithium batteries of fluorinated organic solvents”. In: *Fluorinated Materials for Energy Conversion*. Ed. by Tsuyoshi Nakajima and Henri Groult. Amsterdam: Elsevier Science, 2005, pp. 285–304. ISBN: 978-0-08-044472-7. DOI: <https://doi.org/10.1016/B978-008044472-7/50041-2>.
- [60] Kousuke Hagiya et al. “Physical properties of substituted 1, 3-dioxolan-2-ones”. In: *Chem. Lett.* 37.2 (2008), pp. 210–211.

- [61] Portal Produktowy Grupy PCC Manufacturer of specialty chemicals. *PCC ROKITA Vinylene carbonate (VC)*. URL: https://www.products.pcc.eu/wp-content/uploads/import/broszura/2023-05-24/e3746fad-c86b-4abf-a020-cb20e449f3c4/vinylene-carbonate-vc_broszura_en.pdf. (accessed: 04.05.2024).
- [62] Alar Jänes et al. “Fluoroethylene Carbonate and Propylene Carbonate Mixtures Based Electrolytes for Supercapacitors”. In: *ECS Trans.* 58.27 (2014), p. 71.
- [63] ER Logan et al. “A study of the transport properties of ethylene carbonate-free Li electrolytes”. In: *J. Electrochem. Soc.* 165.3 (2018), A705–A716.
- [64] David S Hall et al. “Exploring classes of co-solvents for fast-charging lithium-ion cells”. In: *J. Electrochem. Soc.* 165.10 (2018), A2365.
- [65] Xiaowei Ma et al. “A study of highly conductive ester co-solvents in Li [Ni0. 5Mn0. 3Co0. 2] O2/Graphite pouch cells”. In: *Electrochim. Acta.* 270 (2018), pp. 215–223.
- [66] Kang Xu. “Navigating the minefield of battery literature”. In: *Commun. Mater.* 3.1 (2022), p. 31.
- [67] Lifei Li et al. “Transport and electrochemical properties and spectral features of non-aqueous electrolytes containing LiFSI in linear carbonate solvents”. In: *J. Electrochem. Soc.* 158.2 (2010), A74.
- [68] Toshihiro Takekawa et al. “Physicochemical and electrochemical properties of the organic solvent electrolyte with lithium bis (fluorosulfonyl) imide (LiFSI) as lithium-ion conducting salt for lithium-ion batteries”. In: *ECS Trans* 64.24 (2015), p. 11.
- [69] Shang Zhu et al. *Differentiable Modeling and Optimization of Battery Electrolyte Mixtures Using Geometric Deep Learning*. 2023. arXiv: 2310.03047 [physics.chem-ph]. URL: <https://arxiv.org/abs/2310.03047>.
- [70] *Methyl acetate*. URL: <http://www.microkat.gr/msdspd90-99/Methyl%20acetate.html>.
- [71] *Methyl propionate*. URL: <https://gestis.dguv.de/data?name=038040&lang=en>.
- [72] *Ethyl formate*. URL: <https://pubchem.ncbi.nlm.nih.gov/compound/Ethyl-formate>.
- [73] Alex K Chew et al. “Advancing material property prediction: using physics-informed machine learning models for viscosity”. In: *Journal of Cheminformatics* 16.1 (2024), p. 31.
- [74] W.M. Haynes. *CRC Handbook of Chemistry and Physics*. CRC Handbook of Chemistry and Physics. CRC Press, 2011. ISBN: 9781439855126.
- [75] *Acetic Anhydride*. URL: https://www.celanese.com/sitecore%20modules/web/-/media/Intermediate%20Chemistry/Files/Product%20Descriptions/Product_Description_and_Handling_Guide-Acetic_Anhydride.pdf.
- [76] *1,4-Dioxane*. URL: <http://www.microkat.gr/msdspd90-99/1%2C4-Dioxane.html>.
- [77] *Perfluoro(4-methylpent-2-ene) TPD-HFPD*. URL: <https://www.fluorochemie.com/product/perfluoro4-methylpent-2-ene-tpd-hfpd>.

- [78] *3M™ Novec™ Introduction 649 Engineered Fluid*. URL: <https://multimedia.3m.com/mws/media/5698650/3m-novec-engineered-fluid-649.pdf>.
- [79] Zhiao Yu et al. “Tuning fluorination of linear carbonate for lithium-ion batteries”. In: *J. Electrochem. Soc.* 169.4 (2022), p. 040555.
- [80] Yan Zhao et al. “Electrolyte engineering via ether solvent fluorination for developing stable non-aqueous lithium metal batteries”. In: *Nat. Commun.* 14.1 (2023), p. 299.
- [81] Zheng Yue et al. “Synthesis and electrochemical properties of partially fluorinated ether solvents for lithiumsulfur battery electrolytes”. In: *J. Power Sources* 401 (2018), pp. 271–277.
- [82] Zhiqiang Zhu et al. “Fluoroethylene Carbonate enabling a robust LiF-rich solid electrolyte interphase to enhance the stability of the MoS₂ Anode for Lithium-ion storage”. In: *Angew. Chem. Int. Ed.* 130.14 (2018), pp. 3718–3722.
- [83] Kihun An et al. “Design of fire-resistant liquid electrolyte formulation for safe and long-cycled lithium-ion batteries”. In: *Adv. Funct. Mater.* 31.48 (2021), p. 2106102.
- [84] Dávid Péter Kovács et al. “MACE-OFF23: Transferable machine learning force fields for organic molecules”. In: *arXiv preprint arXiv:2312.15211* (2023).
- [85] Bowen Deng et al. “CHGNet as a pretrained universal neural network potential for charge-informed atomistic modelling”. In: *Nat. Mach. Intell.* 5.9 (Sept. 2023). DOI: [10.1038/s42256-023-00716-3](https://doi.org/10.1038/s42256-023-00716-3).
- [86] Amil Merchant et al. “Scaling deep learning for materials discovery”. In: *Nature* 624 (Nov. 2023), pp. 1–6. DOI: [10.1038/s41586-023-06735-9](https://doi.org/10.1038/s41586-023-06735-9).
- [87] Duo Zhang et al. *DPA-2: Towards a universal large atomic model for molecular and material simulation*. 2023. arXiv: [2312.15492 \[physics.chem-ph\]](https://arxiv.org/abs/2312.15492).
- [88] Ilyes Batatia et al. *A foundation model for atomistic materials chemistry*. 2024. arXiv: [2401.00096 \[physics.chem-ph\]](https://arxiv.org/abs/2401.00096).
- [89] Dylan Anstine, Roman Zubatyuk, and Olexandr Isayev. *AIMNet2: A Neural Network Potential to Meet your Neutral, Charged, Organic, and Elemental-Organic Needs*. Oct. 2023. DOI: [10.26434/chemrxiv-2023-296ch](https://doi.org/10.26434/chemrxiv-2023-296ch).
- [90] Tianze Zheng et al. “Data-Driven Parametrization of Molecular Mechanics Force Fields for Expansive Chemical Space Coverage”. In: *arXiv preprint arXiv:2408.12817* (2024).
- [91] Samuel P Niblett et al. “Transferability of datasets between Machine-Learning Interaction Potentials”. In: *arXiv preprint arXiv:2409.05590* (2024).
- [92] Naveen Michaud-Agrawal et al. “MDAnalysis: a toolkit for the analysis of molecular dynamics simulations”. In: *J. Comput. Chem.* 32.10 (2011), pp. 2319–2327.
- [93] Xingyu Chen et al. “Polymer architecture effect on sodium ion transport in PSTFSI-based ionomers: A molecular dynamics study”. In: *Solid State Ion.* 288 (2016), pp. 271–276.

- [94] Harish Gudla, Chao Zhang, and Daniel Brandell. “Effects of solvent polarity on Li-ion diffusion in polymer electrolytes: An all-atom molecular dynamics study with charge scaling”. In: *J. Phys. Chem. B.* 124.37 (2020), pp. 8124–8131.
- [95] Aashutosh Mistry et al. “Toward bottom-up understanding of transport in concentrated battery electrolytes”. In: *ACS Cent. Sci.* 8.7 (2022), pp. 880–890.
- [96] R Väli, A Jänes, and E Lust. “Vinylene carbonate as co-solvent for low-temperature mixed electrolyte based Supercapacitors”. In: *J. Electrochem. Soc.* 163.6 (2016), A851.
- [97] US Coast Guard. “Chemical hazard response information system (CHRIS)-hazardous chemical data”. In: *Commandant Instruction* 16465 (1999).
- [98] 3MTM NovecTM 7000 Engineered Fluid. URL: <https://multimedia.3m.com/mws/media/1213720/3m-novec-7000-engineered-fluid-tds.pdf>.
- [99] Adarsh Dave et al. “Autonomous optimization of non-aqueous Li-ion battery electrolytes via robotic experimentation and machine learning coupling”. In: *Nat. Commun.* 13.1 (2022), p. 5454.
- [100] Jianhui Wang et al. “Superconcentrated electrolytes for a high-voltage lithium-ion battery”. In: *Nat. Commun.* 7.1 (2016), p. 12032.
- [101] Noritoshi NAMBU et al. “Use of Monofluorinated Ethyl Propionates as Solvents for Lithium Secondary Batteries”. In: *Electrochemistry* 80.10 (2012), pp. 746–748. doi: [10.5796/electrochemistry.80.746](https://doi.org/10.5796/electrochemistry.80.746).
- [102] John A Dean. *Lange's handbook of chemistry*. 1999.
- [103] Michael S Ding and T Richard Jow. “Conductivity and viscosity of PC-DEC and PC-EC solutions of LiPF₆”. In: *J. Electrochem. Soc.* 150.5 (2003), A620.