

Supporting Information for:
New Features of the Open Source Monte Carlo Software
Brick-CFCMC: Thermodynamic Integration and Hybrid Trial Moves

H. Mert Polat,^{†,‡,¶} Hiran S. Salehi,[‡] Remco Hens,[‡] Dominika O. Wasik,[§] Ahmadreza Rahbari,[‡]
Frédéric de Meyer,^{†,¶} Céline Houriez,[¶] Christophe Coquelet,[¶] Sofia Calero,^{§,||} David Dubbeldam,[⊥]
Othonas A. Moutos,[‡] and Thijs J.H. Vlugt^{*,‡}

[†]*CCUS and Acid Gas Entity, Liquefied Natural Gas Department, Exploration Production,
TotalEnergies S.E., 92078 Paris, France*

[‡]*Engineering Thermodynamics, Process & Energy Department, Faculty of Mechanical, Maritime
and Materials Engineering, Delft University of Technology, Leeghwaterstraat 39, Delft 2628CB,
The Netherlands*

[¶]*CTP - Centre of Thermodynamics of Processes, Mines ParisTech, PSL University, 35 rue Saint
Honoré, 77305 Fontainebleau, France*

[§]*Materials Simulation and Modelling, Department of Applied Physics, Eindhoven University of
Technology, 5600MB Eindhoven, The Netherlands*

^{||}*Department of Physical, Chemical and Natural Systems, Universidad Pablo de Olavide, Ctra.
Utrera Km. 1 Seville ES-41013, Spain*

[⊥]*Van der Hoff Institute for Molecular Sciences, University of Amsterdam, 1098XH Amsterdam,
The Netherlands*

E-mail: t.j.h.vlugt@tudelft.nl

Thermodynamic Integration

In the Continuous Fractional Component Monte Carlo (CFCMC) method, the interactions of the fractional molecule group are scaled using a scaling factor $\lambda^{1,2}$. The fractional molecule group has no interactions with the surrounding molecules when $\lambda = 0$, and has full interactions when $\lambda = 1$. The ensemble average of the derivative of potential energy with respect to λ can be used to calculate the excess chemical potential of species i according to^{3,4}:

$$NVT \text{ ensemble : } \mu_i^{\text{ex}} = \int_0^1 d\lambda \left\langle \frac{\partial A}{\partial \lambda} \right\rangle_{NVT} = \int_0^1 d\lambda \left\langle \frac{\partial U}{\partial \lambda} \right\rangle_{NVT} \quad (\text{S1})$$

$$NPT \text{ ensemble : } \mu_i^{\text{ex}} = \int_0^1 d\lambda \left\langle \frac{\partial G}{\partial \lambda} \right\rangle_{NPT} = \int_0^1 d\lambda \left\langle \frac{\partial U}{\partial \lambda} \right\rangle_{NPT} \quad (\text{S2})$$

Lennard-Jones (LJ) and electrostatic interactions are scaled independently with two different scaling factors, λ_{LJ} and λ_{el} , respectively. Both λ_{LJ} and λ_{el} are a function of λ in such a way that they are both zero when $\lambda = 0$ and both are unity when $\lambda = 1$. For the implementation of the thermodynamic integration, we need to develop a scaling scheme that uses continuous functions λ_{LJ} and λ_{el} so that the integration of Eqs. (S1) and (S2) can be performed. Both λ_{LJ} and λ_{el} and the derivatives of these functions with respect to λ are required to be continuous functions. The following equations provide the used expressions for λ_{LJ} and λ_{el} , respectively:

$$\lambda_{\text{LJ}} = \begin{cases} \frac{20}{9}\lambda & 0.0 < \lambda < 0.4, \\ 1 - \frac{100}{9}(\lambda - \frac{1}{2})^2 & 0.4 < \lambda < 0.5, \\ 1 & 0.5 < \lambda < 1.0 \end{cases} \quad (\text{S3})$$

$$\lambda_{\text{el}} = \begin{cases} 0 & 0.0 < \lambda < 0.5, \\ \frac{100}{9}(\lambda - \frac{1}{2})^2 & 0.5 < \lambda < 0.6, \\ \frac{-11}{9} + \frac{20}{9}\lambda & 0.6 < \lambda < 1.0 \end{cases} \quad (\text{S4})$$

Fig. S1 shows λ_{LJ} and λ_{el} as a function of λ . With this scaling scheme, electrostatic interactions are not “switched on” before the LJ interactions are at full strength ($\lambda_{\text{LJ}} = 1$)⁵. This is chosen to avoid any overlap between the atoms of the fractional group and other atoms. In this way, we protect the electrostatic interaction sites using the LJ interactions in order to avoid atomic overlaps. This scaling scheme can be easily modified in the source code (file: `interactionlambda.f`). It is important to note that both λ_{LJ} and λ_{el} should be continuous functions of λ , and that λ_{el} should be zero when $\lambda_{\text{LJ}} < 1$.

The value of $\frac{\partial U}{\partial \lambda}$ is computed by using the chain rule:

$$\frac{\partial U}{\partial \lambda} = \frac{\partial U_{\text{LJ}}}{\partial \lambda_{\text{LJ}}} \frac{\partial \lambda_{\text{LJ}}}{\partial \lambda} + \frac{\partial U_{\text{el}}}{\partial \lambda_{\text{el}}} \frac{\partial \lambda_{\text{el}}}{\partial \lambda} \quad (\text{S5})$$

The terms $\frac{\partial \lambda_{\text{LJ}}}{\partial \lambda}$ and $\frac{\partial \lambda_{\text{el}}}{\partial \lambda}$ in Eq. (S5) are computed using Eq. (S3) and Eq. (S4). Note that with these definitions, $\frac{\partial U}{\partial \lambda} = 0$ at $\lambda = 0.5$. The terms $\frac{\partial U_{\text{LJ}}}{\partial \lambda_{\text{LJ}}}$ and $\frac{\partial U_{\text{el}}}{\partial \lambda_{\text{el}}}$ are computed after every MC trial move and we keep track of these quantities during the MC simulation. This bookkeeping is implemented to avoid any additional computational cost. In the next two subsections, exact analytic expressions are presented for the computation of $\frac{\partial U_{\text{LJ}}}{\partial \lambda_{\text{LJ}}}$ and $\frac{\partial U_{\text{el}}}{\partial \lambda_{\text{el}}}$.

Lennard-Jones Interactions

Intermolecular Lennard-Jones Interactions

The intermolecular LJ energy between interaction site i of molecule m and interaction site j of molecule n is computed using:

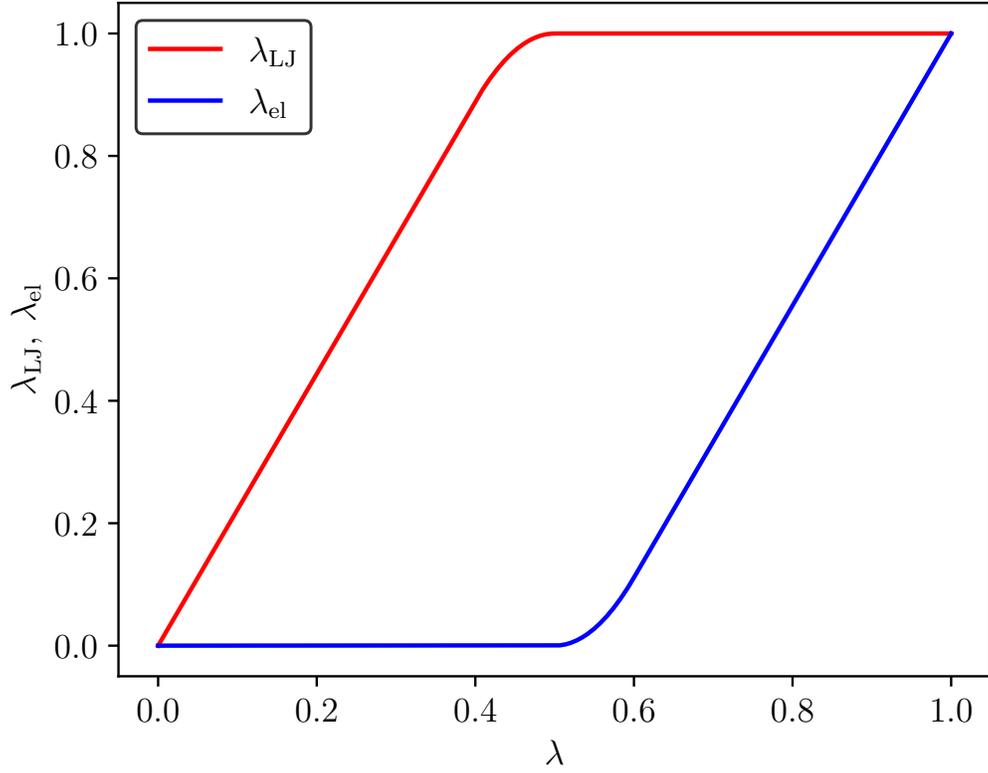


Figure S1: Scaling of LJ and electrostatic interactions as a function of λ for the computation of $\langle \frac{\partial U}{\partial \lambda} \rangle$ which is used for thermodynamic integration in Brick-CFCMC. Red and blue lines are the plots of Eq. (S3) and Eq. (S4), respectively.

$$U_{\text{LJ},ij} = 4\epsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] \quad (\text{S6})$$

where ϵ_{ij} is the minimum of the LJ potential between sites i and j , σ_{ij} is the distance where the LJ potential between sites i and j is zero, and r_{ij} is the distance between sites i and j . When a site belonging to a fractional molecule is involved, the intermolecular LJ energy between site i of molecule m and site j of molecule n is computed using a softcore potential⁶:

$$U_{\text{LJ},ij} = 4\epsilon_{ij}\lambda_{\text{LJ},t} \left[\left(\frac{1}{Y} \right)^{12/c} - \left(\frac{1}{Y} \right)^{6/c} \right] \quad (\text{S7})$$

where the total interaction scaling parameter is computed using the value of λ_{LJ} of sites m

and n ($\lambda_{\text{LJ},t} = \lambda_{\text{LJ},m}\lambda_{\text{LJ},n}$), and

$$Y = \alpha_{\text{LJ}}(1 - \lambda_{\text{LJ},t})^b + (r_{ij}/\sigma_{ij})^c \quad (\text{S8})$$

The values of α_{LJ} , b and c can be adjusted¹. In Brick-CFCMC, the default values for these parameters are 0.5, 1 and 6, respectively. The derivative with respect to $\lambda_{\text{LJ},m}$ can be calculated as:

$$\frac{\partial U_{\text{LJ},ij}}{\partial \lambda_{\text{LJ},m}} = 4\epsilon_{ij}\lambda_{\text{LJ},n} \left(\frac{1}{Y}\right)^{6/c} \left[\left(\frac{1}{Y}\right)^{6/c} - 1 + \frac{6\lambda_{\text{LJ},t}b\alpha_{\text{LJ}}}{cY}(1 - \lambda_{\text{LJ},t})^{b-1} \left(2\left(\frac{1}{Y}\right)^{6/c} - 1 \right) \right] \quad (\text{S9})$$

After every MC trial move, the values of both U and $\frac{\partial U_{\text{LJ},ij}}{\partial \lambda_{\text{LJ},m}}$ are updated.

Tail Corrections

The LJ energy tail correction of system is computed as⁷:

$$U_{\text{LJ}}^{\text{tail}} = \frac{1}{2} \sum_{i,j} \frac{16\pi N_i N_j \epsilon_{ij}}{V} \left(\frac{\sigma_{ij}^{12}}{9r_{\text{cut}}^9} - \frac{\sigma_{ij}^6}{3r_{\text{cut}}^3} \right) \quad (\text{S10})$$

where the sum ranges over all atom types in the system, N_i and N_j are the numbers of atoms of types i and j , respectively (excluding atoms of fractional molecules), and V is the volume of the simulation box. The factor 1/2 accounts for double counting of the interactions.

In principle, there are multiple ways of adding the contribution of the fractional molecule to the tail correction energy. In Brick-CFCMC, this is achieved by substituting $N_i \rightarrow N_i + \lambda_{\text{LJ},m} \times N_{m,i}$ and $N_j \rightarrow N_j + \lambda_{\text{LJ},n} \times N_{n,j}$ in Eq. (S10), where $N_{m,i}$ and $N_{n,j}$ are the

numbers of atoms of types i and j within fractional molecules m and n , respectively⁷. The derivative with respect to $\lambda_{\text{LJ},m}$ can be calculated as:

$$\frac{\partial U_{\text{LJ}}^{\text{tail}}}{\partial \lambda_{\text{LJ},m}} = \frac{1}{2V} \sum_{i,j} \left[16\pi\epsilon_{ij}(N_j N_{m,i} + \lambda_{\text{LJ},n} N_{m,i} N_{n,j}) \left[\left(\frac{\sigma_{ij}^{12}}{9r_{\text{cut}}^9} \right) - \left(\frac{\sigma_{ij}^6}{3r_{\text{cut}}^3} \right) \right] \right] \quad (\text{S11})$$

Electrostatic Interactions

The analytic expressions for $\frac{\partial U_{\text{el}}}{\partial \lambda_{\text{el}}}$ of electrostatic potentials may seem trivial at first sight because for linear charge scaling $\frac{\partial U_{\text{el}}}{\partial \lambda_{\text{el}}}$ is proportional to λ_{el} ^{8,9}. It is important to note that such a scaling may result in numerical instabilities and atomic overlaps. The electrostatic interaction potentials for the fractional molecule groups are defined in Brick-CFCMC in such a way that they have an offset parameter (Q) to avoid any atomic overlaps (e.g. see Eq. (S13)). Therefore, the computation of $\frac{\partial U_{\text{el}}}{\partial \lambda_{\text{el}}}$ will require more complex expressions. The next three subsections present the analytic expressions of $\frac{\partial U_{\text{el}}}{\partial \lambda_{\text{el}}}$ for the Ewald summation¹⁰, the Wolf method¹¹, and the damped and shifted version of the Wolf method¹².

Ewald Summation

The Ewald summation consists of a real-space part, exclusion part, self-energy part, and Fourier-space part^{10,13,14}. The real-space part is a damped electrostatic potential for the short-ranged interactions. The exclusion accounts for all intramolecular interactions for which electrostatic interactions should not be considered (i.e. between atoms that interact with a bonded interaction potential such as bond-stretching or bond-bending). The self-energy part considers the self-electrostatic energy of all charges, and the Fourier-space part handles the long-range electrostatic interactions by using a Fourier transform. The real-space energy of the Ewald summation between sites i and j of molecules m and n is computed as:

$$U_{\text{real},ij} = q_i q_j \frac{\text{erfc}(\alpha_{\text{el}} r_{ij})}{r_{ij}} \quad (\text{S12})$$

where α_{el} is the damping parameter, $\text{erfc}(x)$ is the complementary error function and r_{ij} is the distance between the interaction sites i and j . If one of the sites involved in the interaction belongs to a fractional molecule group, then the real-space energy of the Ewald summation is computed as:

$$U_{\text{real},ij} = \lambda_{\text{el,t}} q_i q_j \frac{\text{erfc}(\alpha_{\text{el}}(r_{ij} + Q))}{r_{ij} + Q} \quad (\text{S13})$$

where the total fractional scaling factor for electrostatic interactions is computed by multiplying the electrostatic interaction scaling factors of molecules m and n ($\lambda_{\text{el,t}} = \lambda_{\text{el,m}} \lambda_{\text{el,n}}$). Q is the offset parameter computed as $Q = \beta_{\text{el}}(1 - \lambda_{\text{el,t}})$ where β_{el} is equal to 0.01 Å. In this way, there is no divergence of the interaction potential, even when $r_{ij} = 0$. The exclusion term of the Ewald summation between atoms i and j in molecule m is obtained using^{15,16}:

$$U_{\text{exclusion},ij} = q_i q_j \frac{\text{erfc}(\alpha_{\text{el}} r_{ij}) - 1}{r_{ij}} \quad (\text{S14})$$

When a fractional molecule is involved, the exclusion term is computed as follows:

$$U_{\text{exclusion},ij} = \lambda_{\text{el,t}} q_i q_j \frac{\text{erfc}(\alpha_{\text{el}}(r_{ij} + Q)) - 1}{r_{ij} + Q} \quad (\text{S15})$$

The self-energy term of the Ewald summation is computed as:

$$U_{\text{self}} = \frac{-\alpha_{\text{el}}}{\sqrt{\pi}} \sum_n \sum_i \lambda_{\text{el,n}}^2 q_{i,n}^2 \quad (\text{S16})$$

where index n runs over all the molecules in the simulation box and index i runs over all

atoms in molecule n , and $\lambda_{\text{el},t} = \lambda_{\text{el},m}^2$. The Fourier-space term of the Ewald summation is computed as¹⁰:

$$U_{\text{Fourier}} = \frac{1}{V} \sum_{\mathbf{k}} F(\mathbf{k}) \left[\left(\sum_i \lambda_{\text{el},i} q_i \cos(\mathbf{i}\mathbf{k} \cdot \mathbf{r}_i) \right)^2 + \left(\sum_i \lambda_{\text{el},i} q_i \sin(\mathbf{i}\mathbf{k} \cdot \mathbf{r}_i) \right)^2 \right] \quad (\text{S17})$$

in which indices \mathbf{k} and i run over all k -vectors (except for the zero wavevector $\mathbf{k} = (0, 0, 0)$)³ and all atoms in the system, respectively, and:

$$F(\mathbf{k}) = \frac{4\pi}{|\mathbf{k}|^2} \exp \left[\frac{|\mathbf{k}|^2}{4\alpha_{\text{el}}} \right] \quad (\text{S18})$$

In Eq. (S17), the terms $\lambda_{\text{el},i}$ are the electrostatic interaction scaling factors of atoms i . These terms were added to this equation to account for the contribution of the fractional molecules to the Fourier-space part of the Ewald summation. The values of $\lambda_{\text{el},i}$ are set to 1 for atoms of whole molecules (i.e. molecules that always have a full interaction strength with the surrounding molecules⁷) and for site belonging to fractional molecules, $\lambda_{\text{el},i} = \lambda_{\text{el}}$.

The bookkeeping for the real-space and exclusion parts of the Ewald summation is similar to the bookkeeping of the LJ interactions as these are all pairwise interactions³. The bookkeeping of the self-energy term is trivial as this term does not depend on the atomic positions. The bookkeeping of the Fourier-space part of the Ewald summation is more complicated as it is not a pairwise interaction. In Brick-CFCMC, the bookkeeping for the Fourier-space term of the Ewald summation is performed according to Ref.¹³. During the simulation, we keep track of the values of the terms in $\sum_i \lambda_{\text{el},i} q_i \cos(\mathbf{i}\mathbf{k} \cdot \mathbf{r}_i)$ and $\sum_i \lambda_{\text{el},i} q_i \sin(\mathbf{i}\mathbf{k} \cdot \mathbf{r}_i)$ of Eq. (S17). For each wavevector \mathbf{k} , this requires the storage of two floats. After every MC trial move, the values of the terms in $\sum_i \lambda_{\text{el},i} q_i \cos(\mathbf{i}\mathbf{k} \cdot \mathbf{r}_i)$ and $\sum_i \lambda_{\text{el},i} q_i \sin(\mathbf{i}\mathbf{k} \cdot \mathbf{r}_i)$ are updated by adding the contribution of the new configuration and subtracting the contribution of the old configuration. This updating is performed for each wavevector separately.

The derivative of the real-space part of the Ewald summation with respect to the scaling factor of (fractional) molecule m $\lambda_{\text{el},m}$ is obtained as:

$$\frac{\partial U_{\text{real},ij}}{\partial \lambda_{\text{el},m}} = \lambda_n q_i q_j \left[\frac{\text{erfc}(\alpha_{\text{el}}(r_{ij} + Q))}{r_{ij} + Q} + \lambda_{\text{el}}(X(r_{ij})) \right] \quad (\text{S19})$$

where $X(r_{ij})$ is computed using:

$$X(r) = \frac{\beta_{\text{el}}}{(r + Q)^2} \left[(r + Q) \left(\frac{2\alpha_{\text{el}}}{\sqrt{\pi}} \right) \exp[-\alpha_{\text{el}}^2(r + Q)^2] + \text{erfc}(\alpha_{\text{el}}(r + Q)) \right] \quad (\text{S20})$$

The derivative of the exclusion part of the Ewald summation with respect to $\lambda_{\text{el},m}$ is calculated according to:

$$\frac{\partial U_{\text{exclusion},ij}}{\partial \lambda_{\text{el},m}} = 2\lambda_{\text{el},m} q_i q_j \left[\frac{\text{erfc}(\alpha_{\text{el}}(r_{ij} + Q))}{r_{ij} + Q} + \lambda_{\text{el},t}(X(r_{ij})) - \frac{r_{ij} + Q + \lambda_{\text{el},t}\beta_{\text{el}}}{(r_{ij} + Q)^2} \right] \quad (\text{S21})$$

where $X(r)$ follows Eq. (S20). The derivative of the self-energy part with respect to $\lambda_{\text{el},m}$ of fractional molecule m becomes:

$$\frac{\partial U_{\text{self}}}{\partial \lambda_{\text{el},m}} = 2\lambda_{\text{el},m} \frac{-\alpha_{\text{el}}}{\sqrt{\pi}} \sum_j q_j^2 \quad (\text{S22})$$

in which index j runs over all atoms in the fractional molecule m . The derivative of the Fourier-space part of the Ewald summation with respect to $\lambda_{\text{el},m}$ is:

$$\frac{\partial U_{\text{Fourier}}}{\partial \lambda_{\text{el},m}} = \frac{2}{V} \sum_{\mathbf{k}} F(\mathbf{k}) \left[\left(\sum_i \lambda_{\text{el},i} q_i \cos(\mathbf{i}\mathbf{k} \cdot \mathbf{r}_i) \right) \times \left(\sum_j q_j \cos(\mathbf{i}\mathbf{k} \cdot \mathbf{r}_j) \right) + \left(\sum_i \lambda_{\text{el},i} q_i \sin(\mathbf{i}\mathbf{k} \cdot \mathbf{r}_i) \right) \times \left(\sum_j q_j \sin(\mathbf{i}\mathbf{k} \cdot \mathbf{r}_j) \right) \right] \quad (\text{S23})$$

in which indices k , i and j run over all k -vectors (except for the zero wavevector $\mathbf{k} = (0, 0, 0)$), all atoms in the system and all atoms of molecule m , respectively. $F(\mathbf{k})$ follows Eq. (S18) and V is the volume of the simulation box. The bookkeeping for the computation of $\frac{\partial U_{\text{Fourier}}}{\partial \lambda_{\text{el},m}}$ is performed in the same way as the bookkeeping for the Fourier-space energy of the Ewald summation. The values of the terms in $(\sum_i \lambda_{\text{el},i} q_i \cos(\mathbf{i}\mathbf{k} \cdot \mathbf{r}_i)) \times (\sum_j q_j \cos(\mathbf{i}\mathbf{k} \cdot \mathbf{r}_j))$ and $(\sum_i \lambda_{\text{el},i} q_i \sin(\mathbf{i}\mathbf{k} \cdot \mathbf{r}_i)) \times (\sum_j q_j \sin(\mathbf{i}\mathbf{k} \cdot \mathbf{r}_j))$ are calculated at the start of the simulation and stored in memory. The values of the terms in $(\sum_i \lambda_{\text{el},i} q_i \cos(\mathbf{i}\mathbf{k} \cdot \mathbf{r}_i)) \times (\sum_j q_j \cos(\mathbf{i}\mathbf{k} \cdot \mathbf{r}_j))$ and $(\sum_i \lambda_{\text{el},i} q_i \sin(\mathbf{i}\mathbf{k} \cdot \mathbf{r}_i)) \times (\sum_j q_j \sin(\mathbf{i}\mathbf{k} \cdot \mathbf{r}_j))$ are updated after every MC trial move by subtracting the contribution of the old configuration and adding the contribution of the new configuration. This is done for each wavevector. Because this is performed only for the atoms that have different positions in the old and the new configurations, this does not lead to any additional computational cost.

Wolf Method

The Wolf method uses the strong screening of the electrostatic interactions in a system to calculate electrostatic potential energy¹¹. Because of this strong screening, it works very well for dense (liquid) systems^{17,18} while it does not work well for the less dense (gas) systems due to less effective screening of electrostatics^{13,15}. All interactions in the Wolf method are either pairwise interactions (real-space and exclusion parts) or constant (self-energy part) and no Fourier transform is involved. This makes this method computationally more efficient than the Ewald summation¹⁵. The short ranged real-space electrostatic energy between site i of

molecule m and site j of molecule n is computed as:

$$U_{\text{real},ij} = q_i q_j \left[\frac{\text{erfc}(\alpha_{\text{el}} r_{ij})}{r_{ij}} - \frac{\text{erfc}(\alpha_{\text{el}} r_{\text{cut}})}{r_{\text{cut}}} \right] \quad (\text{S24})$$

where $\text{erfc}(x)$ is the complementary error function, r_{ij} is the distance between sites i and j , r_{cut} is the cutoff radius, and α_{el} is the damping parameter. When a site of a fractional molecule group is involved in this type of interaction, the following expression is used to compute the real-space electrostatic energy between site i of molecule m and site j of molecule n .

$$U_{\text{real},ij} = \lambda_{\text{el},t} q_i q_j \left[\frac{\text{erfc}(\alpha_{\text{el}}(r_{ij} + Q))}{r_{ij} + Q} - \frac{\text{erfc}(\alpha_{\text{el}}(r_{\text{cut}} + Q))}{r_{\text{cut}} + Q} \right] \quad (\text{S25})$$

in which $\lambda_{\text{el},t} = \lambda_{\text{el},m} \lambda_{\text{el},n}$, and the offset term Q is:

$$Q = \beta_{\text{el}}(1 - \lambda_{\text{el},t}) \quad (\text{S26})$$

where β_{el} is taken as 0.01 Å. The derivative of the real-space part of the Wolf method with respect to $\lambda_{\text{el},m}$ is obtained as:

$$\frac{\partial U_{\text{real},ij}}{\partial \lambda_{\text{el},m}} = \lambda_n q_i q_j \left[\frac{\text{erfc}(\alpha_{\text{el}}(r_{ij} + Q))}{r_{ij} + Q} - \frac{\text{erfc}(\alpha_{\text{el}}(r_{\text{cut}} + Q))}{r_{\text{cut}} + Q} + \lambda_{\text{el},t} (X(r_{ij}) - X(r_{\text{cut}})) \right] \quad (\text{S27})$$

where the term $X(r)$ follows from Eq. (S20).

The exclusion term of the Wolf method between sites i and j of molecule m is computed as:

$$U_{\text{exclusion},ij} = q_i q_j \left[\frac{\text{erfc}(\alpha_{\text{el}} r_{ij}) - 1}{r_{ij}} - \frac{\text{erfc}(\alpha_{\text{el}} r_{\text{cut}})}{r_{\text{cut}}} \right] \quad (\text{S28})$$

When molecule m is in a fractional molecule group, the following expression is used for the exclusion term:

$$U_{\text{exclusion},ij} = \lambda_{\text{el},t} q_i q_j \left[\frac{\text{erfc}(\alpha_{\text{el}}(r_{ij} + Q)) - 1}{r_{ij} + Q} - \frac{\text{erfc}(\alpha_{\text{el}}(r_{\text{cut}} + Q))}{r_{\text{cut}} + Q} \right] \quad (\text{S29})$$

in which $\lambda_{\text{el,t}} = \lambda_{\text{el,m}}^2$, and the derivative with respect to $\lambda_{\text{el,m}}$ is calculated from:

$$\frac{\partial U_{\text{exclusion,ij}}}{\partial \lambda_{\text{el,m}}} = 2\lambda_{\text{el,m}}q_iq_j \left[\frac{\text{erfc}(\alpha_{\text{el}}(r_{ij} + Q))}{r_{ij} + Q} - \frac{\text{erfc}(\alpha_{\text{el}}(r_{\text{cut}} + Q))}{r_{\text{cut}} + Q} + \lambda_{\text{el,t}}(X(r_{ij}) - X(r_{\text{cut}})) - \frac{r_{ij} + Q + \lambda_{\text{el}}\beta_{\text{el}}}{(r_{ij} + Q)^2} \right] \quad (\text{S30})$$

The self-energy term of the Wolf method is computed from:

$$U_{\text{self}} = - \left(\frac{\text{erfc}(\alpha_{\text{el}}r_{\text{cut}})}{2r_{\text{cut}}} + \frac{\alpha_{\text{el}}}{\sqrt{\pi}} \right) \sum_n \sum_i \lambda_{\text{el,n}}^2 q_{i,n}^2 \quad (\text{S31})$$

where index n runs over all molecules in the simulation box and index i runs over all atoms in molecule n . The derivative for the self-energy term of the Wolf method is computed as follows:

$$\frac{\partial U_{\text{self}}}{\partial \lambda_{\text{el,m}}} = -2\lambda_{\text{el,m}} \left(\frac{\text{erfc}(\alpha_{\text{el}}r_{\text{cut}})}{2r_{\text{cut}}} + \frac{\alpha_{\text{el}}}{\sqrt{\pi}} \right) \sum_j q_j^2 \quad (\text{S32})$$

where index j runs over all atoms in fractional molecule m .

Damped and Shifted Version of the Wolf Method

The Wolf method can be used to accurately calculate electrostatic interactions, however, artificial structuring around the cutoff distance is a potential problem¹⁹. Fennell and Gezelter¹² solved this issue with a modification to the real-space term of the Wolf method. The real-space electrostatic energy between site i of molecule m and site j of molecule n in the damped and shifted (DSF) version of the Wolf method is computed as^{15,16}:

$$U_{\text{real},ij} = q_i q_j \left[\frac{\text{erfc}(\alpha_{\text{el}} r_{ij})}{r_{ij}} - \frac{\text{erfc}(\alpha_{\text{el}} r_{\text{cut}})}{r_{\text{cut}}} + \left(\frac{\text{erfc}(\alpha_{\text{el}} r_{\text{cut}})}{r_{\text{cut}}^2} + \frac{2\alpha_{\text{el}} \exp[-\alpha_{\text{el}}^2 r_{\text{cut}}^2]}{\sqrt{\pi} r_{\text{cut}}} (r_{ij} - r_{\text{cut}}) \right) \right] \quad (\text{S33})$$

where $\text{erfc}(x)$ is the complementary error function, r_{ij} is the distance between the sites i and j , r_{cut} is the cutoff radius, and α_{el} is the damping parameter. In this way, both $U_{\text{real},ij}$ and its derivative with respect to r_{ij} are continuous around r_{cut} . When either one or both of the sites involved in the interaction belong to a fractional molecule group, the real-space term in DSF Wolf method is computed from:

$$U_{\text{real},ij} = \lambda_{\text{el},t} q_i q_j \left[\frac{\text{erfc}(\alpha_{\text{el}}(r_{ij} + Q))}{r_{ij} + Q} - \frac{\text{erfc}(\alpha_{\text{el}}(r_{\text{cut}} + Q))}{r_{\text{cut}} + Q} + \left(\frac{\text{erfc}(\alpha_{\text{el}} r_{\text{cut}})}{r_{\text{cut}}^2} + \frac{2\alpha_{\text{el}} \exp[-\alpha_{\text{el}}^2 r_{\text{cut}}^2]}{\sqrt{\pi} r_{\text{cut}}} (r_{ij} - r_{\text{cut}}) \right) \right] \quad (\text{S34})$$

where the term Q follows from Eq. (S26). The total scaling factor for electrostatic interactions $\lambda_{\text{el},t}$ is calculated using the value of λ_{el} of the molecules m and n as $\lambda_{\text{el},t} = \lambda_{\text{el},m} \lambda_{\text{el},n}$. The derivative of the DSF real-space term with respect to λ_m is calculated using:

$$\frac{\partial U_{\text{real},ij}^{\text{DSF}}}{\partial \lambda_{\text{el},m}} = \lambda_{\text{el},n} q_i q_j \left[\frac{\text{erfc}(\alpha_{\text{el}}(r_{ij} + \beta_{\text{el}}(1 - \lambda_{\text{el},t})))}{r_{ij} + \beta_{\text{el}}(1 - \lambda_{\text{el},t})} - \frac{\text{erfc}(\alpha_{\text{el}}(r_{\text{cut}} + \beta_{\text{el}}(1 - \lambda_{\text{el},t})))}{r_{\text{cut}} + \beta_{\text{el}}(1 - \lambda_{\text{el},t})} + \left(\frac{\text{erfc}(\alpha_{\text{el}} r_{\text{cut}})}{r_{\text{cut}}^2} + \frac{2\alpha_{\text{el}} \exp[-\alpha_{\text{el}}^2 r_{\text{cut}}^2]}{\sqrt{\pi} r_{\text{cut}}} \right) (r_{ij} - r_{\text{cut}}) + \lambda_{\text{el},t} (X(r_{ij}) - X(r_{\text{cut}})) \right] \quad (\text{S35})$$

where the term $X(r)$ follows from Eq. (S20). The exclusion term of the DSF version of the Wolf method between sites i and j of the molecule m is computed using:

$$U_{\text{exclusion},ij} = q_i q_j \left[\frac{\text{erfc}(\alpha_{\text{el}} r_{ij}) - 1}{r_{ij}} - \frac{\text{erfc}(\alpha_{\text{el}} r_{\text{cut}})}{r_{\text{cut}}} \right] \quad (\text{S36})$$

The exclusion term of DSF version of the Wolf method between sites i and j of molecule m ,

when m is a fractional molecule, is computed as:

$$U_{\text{exclusion},ij} = \lambda_{\text{el},t} q_i q_j \left[\frac{\text{erfc}(\alpha_{\text{el}}(r_{ij} + Q)) - 1}{r_{ij} + Q} - \frac{\text{erfc}(\alpha_{\text{el}}(r_{\text{cut}} + Q))}{r_{\text{cut}} + Q} \right] \quad (\text{S37})$$

where $\lambda_{\text{el},t} = \lambda_{\text{el},m}^2$. The derivative of the DSF exclusion term with respect to $\lambda_{\text{el},m}$ is calculated from:

$$\frac{\partial U_{\text{exclusion},ij}}{\partial \lambda_{\text{el},m}} = 2\lambda_{\text{el},m} q_i q_j \left[\frac{\text{erfc}(\alpha_{\text{el}}(r_{ij} + Q))}{r_{ij} + Q} - \frac{\text{erfc}(\alpha_{\text{el}}(r_{\text{cut}} + Q))}{r_{\text{cut}} + Q} + \lambda_{\text{el},t} (X(r_{ij}) - X(r_{\text{cut}})) - \frac{r_{ij} + Q + \lambda_{\text{el}}\beta_{\text{el}}}{(r_{ij} + Q)^2} \right] \quad (\text{S38})$$

where $X(r)$ follows from Eq. (S20). The self-energy term of DSF version of the Wolf method is computed as:

$$U_{\text{self}} = - \left(\frac{\text{erfc}(\alpha_{\text{el}} r_{\text{cut}})}{2r_{\text{cut}}} + \frac{\alpha_{\text{el}}}{\sqrt{\pi}} \right) \sum_n \sum_i \lambda_{n,\text{el}}^2 q_{i,n}^2 \quad (\text{S39})$$

in which n runs over all molecules in the system and i runs over all atoms in molecule n . The derivative of the self-energy term with respect to $\lambda_{\text{el},m}$ of fractional molecule m becomes:

$$\frac{\partial U_{\text{self}}}{\partial \lambda_{\text{el},m}} = -2\lambda_{\text{el},m} \left(\frac{\text{erfc}(\alpha_{\text{el}} r_{\text{cut}})}{2r_{\text{cut}}} + \frac{\alpha_{\text{el}}}{\sqrt{\pi}} \right) \sum_j q_j^2 \quad (\text{S40})$$

in which index j runs over all atoms in fractional molecule m .

Hybrid Monte Carlo Trial Moves

The hybrid trial moves use a short Molecular Dynamics (MD) trajectory to simultaneously displace or rotate all molecules inside the simulation box. These trial moves are more efficient than the single-molecule trial moves in inducing a collective motion in the fluid²⁰⁻²². The interaction potential to generate the short MD trajectories does not need to be the actual interaction potential. It can be another interaction potential still resembling the actual one

but computationally cheaper^{3,20}.

Hybrid Translation

In the hybrid MC translation trial move, a short MD simulation in the NVE ensemble is performed with a specified time step (Δt) and trajectory length (N_{step}). Although there are no restrictions on the choice of Δt and N_{step} , these parameters influence the efficiency of the sampling. It is therefore recommended that the optimal values of these parameters are chosen from short test runs, such that an average acceptance probability of ca. 50% and a maximum average displacement per unit of CPU time are achieved. All molecules are kept rigid, and the trial move is performed collectively (for all molecules) using the center of mass motion. To integrate the equations of motion, the velocity Verlet algorithm^{23,24} is used, which is time reversible and area-preserving (symplectic)³. This trial move is performed as follows^{3,24-26}:

1. Center of mass velocity vectors, \mathbf{v}_i , are randomly generated for each molecule i , where the vector components are drawn from a normal distribution with the mean value equal to 0 and a variance of 1.
2. The kinetic energy of translation of the old configuration is calculated as:

$$K_{\text{old}}^{\text{trans}} = \sum_{i=1}^N \frac{1}{2} m_i \mathbf{v}_{i,\text{old}}^2 \quad (\text{S41})$$

where i is the molecule number, N is the total number of molecules in the system, and m_i is the mass of molecule i .

3. All velocities are scaled by the factor $\sqrt{3Nk_{\text{B}}T/2K_{\text{old}}^{\text{trans}}}$ which adjusts the kinetic energy of the system according to the equipartition theorem²¹. The kinetic energy of the system ($K_{\text{old}}^{\text{trans}}$) is then recomputed using Eq. (S41).

4. The resultant force, \mathbf{F}_i , acting on each molecule i is computed. To reduce the computational costs, an approximation can be used for \mathbf{F}_i (rather than its precise value), e.g., obtained from a computationally cheaper method, e.g., the damped and shifted version of the Wolf method instead of the Ewald summation. In principle, such an approximation would maintain a correct phase-space sampling³.

5. The velocities of all molecules are updated to a half time step:

$$\mathbf{v}_{i,\text{old}} = \mathbf{v}_{i,\text{old}} + \frac{\mathbf{F}_i}{2m_i} \Delta t \quad (\text{S42})$$

6. The center-of-mass position vectors of all molecules are updated:

$$\mathbf{r}_{i,\text{new}} = \mathbf{r}_{i,\text{old}} + \mathbf{v}_{i,\text{old}} \Delta t \quad (\text{S43})$$

7. Using the new center-of-mass positions ($\mathbf{r}_{i,\text{new}}$), all atomic coordinates are updated.

8. The resultant forces on all molecules are recomputed based on the new atomic positions $\mathbf{r}_{i,\text{new}}$.

9. The velocities are updated to the full time step:

$$\mathbf{v}_{i,\text{new}} = \mathbf{v}_{i,\text{old}} + \frac{\mathbf{F}_i}{2m_i} \Delta t \quad (\text{S44})$$

10. For an MD trajectory of length N_{step} , steps 5-9 are repeated for $N_{\text{step}} - 1$ times.

11. The kinetic energy of the new configuration is calculated as:

$$K_{\text{new}}^{\text{trans}} = \sum_{i=1}^N \frac{1}{2} m_i \mathbf{v}_{i,\text{new}}^2 \quad (\text{S45})$$

12. The trial move is accepted or rejected according to the following acceptance rule³:

$$\text{acc}(o \rightarrow n) = \min(1, \exp[-\beta(\Delta U + \Delta K^{\text{trans}})]) \quad (\text{S46})$$

where o and n denote the old and new (initial and final) configurations on the MD trajectory, and ΔU and ΔK^{trans} are the differences in potential energy and translational kinetic energy, respectively, between the old and new configurations. β is defined as $1/(k_{\text{B}}T)$, where k_{B} is the Boltzmann constant, and T is the absolute temperature.

Hybrid Rotation

In the hybrid rotation trial move, collective rotation of molecules as rigid bodies is performed using a short MD simulation in the NVE ensemble. The time step size (Δt) and trajectory length (N_{step}) of this MD run are chosen to maximize the efficiency of the sampling. For the rigid body rotation of molecules, the velocity Verlet-based algorithm of Miller et al.²⁷ (NOSQUISH) is used, which is symplectic and time reversible. At every time step, all molecules are rotated according to the total torque acting on the molecules, and only intermolecular interactions are taken into account to compute the forces and torques.

In the first step, the moment of inertia tensor, \mathbf{I}_i , of each molecule i about its center of mass is computed. To obtain the principal moments of inertia, the eigenvalues and eigenvectors of the inertia tensors are computed using the Jacobi method²⁸. The computed eigenvectors indicate the direction of the principal axes, and the corresponding eigenvalues determine the values of inertia moments in these directions. The body frame of reference is taken to be the principal axes of each molecule, denoted by \hat{x} , \hat{y} , and \hat{z} , whereas the laboratory frame axes are specified by x , y , and z . The principal (diagonalized) moment of inertia tensor of molecule i is denoted by $\hat{\mathbf{I}}_i$, and its diagonal elements (eigenvalues of the \mathbf{I}_i tensor), are represented by $\hat{I}_{\hat{x}\hat{x},i}$, $\hat{I}_{\hat{y}\hat{y},i}$, and $\hat{I}_{\hat{z}\hat{z},i}$. The principal axes are set such that $I_{\hat{x}\hat{x},i} > I_{\hat{y}\hat{y},i} > I_{\hat{z}\hat{z},i}$. The quaternion 4-vector, $\mathbf{q}_i^{(4)} = (q_{0,i} \ q_{1,i} \ q_{2,i} \ q_{3,i})^{\text{T}}$ is then computed for

the body frame of each molecule i , based on the direction of the principal axes. Subsequently, the rotation matrix, \mathbf{R}_i , for each molecule i is calculated according to:

$$\mathbf{R}_i = \begin{pmatrix} q_{0,i}^2 + q_{1,i}^2 - q_{2,i}^2 - q_{3,i}^2 & 2(q_{1,i}q_{2,i} - q_{0,i}q_{3,i}) & 2(q_{1,i}q_{3,i} + q_{0,i}q_{2,i}) \\ 2(q_{1,i}q_{2,i} + q_{0,i}q_{3,i}) & q_{0,i}^2 - q_{1,i}^2 + q_{2,i}^2 - q_{3,i}^2 & 2(q_{2,i}q_{3,i} - q_{0,i}q_{1,i}) \\ 2(q_{1,i}q_{3,i} - q_{0,i}q_{2,i}) & 2(q_{2,i}q_{3,i} + q_{0,i}q_{1,i}) & q_{0,i}^2 - q_{1,i}^2 - q_{2,i}^2 + q_{3,i}^2 \end{pmatrix} \quad (\text{S47})$$

In the algorithm outlined here, a variable X computed in the body frame is denoted by \hat{X} , whereas it lacks the hat symbol when designated in the laboratory frame. The following steps are performed in the hybrid rotation trial move^{26,27,29}:

1. Angular velocity vectors, $\hat{\boldsymbol{\omega}}_i$, are randomly generated for every molecule i in its body frame, where each component of the angular velocity ($\omega_{\hat{x},i}$, $\omega_{\hat{y},i}$, and $\omega_{\hat{z},i}$) is obtained from a normal distribution with mean and variance values of 0 and 1, respectively.
2. The rotational kinetic energy of the system in the old configuration is computed according to:

$$K_{\text{old}}^{\text{rot}} = \sum_{i=1}^N \frac{1}{2} (I_{\hat{x}\hat{x},i} \hat{\omega}_{\hat{x},i}^2 + I_{\hat{y}\hat{y},i} \hat{\omega}_{\hat{y},i}^2 + I_{\hat{z}\hat{z},i} \hat{\omega}_{\hat{z},i}^2) \quad (\text{S48})$$

where i is the molecule number, and N is the total number of molecules in the system.

3. Similar to the hybrid translation trial move, the angular velocities of all molecules are scaled by a factor of $\sqrt{3Nk_{\text{B}}T/2K_{\text{old}}^{\text{rot}}}$ to yield the correct temperature and kinetic energy. The rotational kinetic energy ($K_{\text{old}}^{\text{rot}}$) is then recomputed using Eq. (S48).
4. The positions of each atom j in molecule i , with respect to the center-of-mass of i , are computed in the body frame, using the inverse of the rotation matrix of molecule i :

$$\hat{\mathbf{r}}_{ji} = \mathbf{R}_i^{-1} \mathbf{r}_{ji} \quad (\text{S49})$$

where \mathbf{r}_{ji} is the position vector of atom j in molecule i , with respect to the center-of-

mass of i , in the laboratory frame.

5. The resultant torque vector on each molecule i with N_i atoms, \mathbf{T}_i , is calculated about its center of mass in the laboratory frame:

$$\mathbf{T}_i = \sum_{j=1}^{N_i} \sum_k [\mathbf{r}_{ji} \times \mathbf{F}_{jk}] \quad (\text{S50})$$

in which the indices j and k run over all atoms in molecule i and atoms in all other molecules except i , respectively. \mathbf{r}_{ji} is the position vector of atom j with respect to the center-of-mass of molecule i , and \mathbf{F}_{jk} denotes the force vector acting upon atom j of molecule i by atom k of another molecule. An approximate value of the force can be used in Eq. (S50) to reduce the computational costs.

6. The resultant torque on each molecule i in its body frame, $\hat{\mathbf{T}}_i$, is computed using the transposed rotation matrix:

$$\hat{\mathbf{T}}_i = \mathbf{R}_i^T \mathbf{T}_i \quad (\text{S51})$$

7. The 4-vector quaternion torque of each molecule i in its body frame, $\hat{\mathbf{T}}_{\mathbf{q},i}^{(4)}$, is computed according to:

$$\hat{\mathbf{T}}_{\mathbf{q},i}^{(4)} = 2\mathbf{M}_i \hat{\mathbf{T}}_i^{(4)} \quad (\text{S52})$$

where $\hat{\mathbf{T}}_i^{(4)} = (0 \ \hat{T}_{\hat{x},i} \ \hat{T}_{\hat{y},i} \ \hat{T}_{\hat{z},i})^T$ is the 4-vector torque on molecule i in its body frame, and $\hat{T}_{\hat{x},i}$, $\hat{T}_{\hat{y},i}$, and $\hat{T}_{\hat{z},i}$ as the components of the $\hat{\mathbf{T}}_i$ vector. \mathbf{M}_i is a matrix consisting of the quaternion components of the body frame of molecule i :

$$\mathbf{M}_i = \begin{pmatrix} q_{0,i} & -q_{1,i} & -q_{2,i} & -q_{3,i} \\ q_{1,i} & q_{0,i} & -q_{3,i} & q_{2,i} \\ q_{2,i} & q_{3,i} & q_{0,i} & -q_{1,i} \\ q_{3,i} & -q_{2,i} & q_{1,i} & q_{0,i} \end{pmatrix}$$

8. The angular momentum of every molecule i , $\hat{\mathbf{L}}_i$ is computed in its body frame using the principal moment of inertia tensor and the angular velocity:

$$\hat{\mathbf{L}}_i = \hat{\mathbf{I}}_i \hat{\boldsymbol{\omega}}_i \quad (\text{S53})$$

9. The 4-vector conjugate quaternion momentum of each molecule i , $\mathbf{P}_{\mathbf{q},i}^{(4)}$, is computed as:

$$\mathbf{P}_{\mathbf{q},i}^{(4)} = 2\mathbf{M}_i \hat{\mathbf{L}}_i^{(4)} \quad (\text{S54})$$

where $\hat{\mathbf{L}}_i^{(4)} = (0 \ \hat{L}_{\hat{x},i} \ \hat{L}_{\hat{y},i} \ \hat{L}_{\hat{z},i})^T$ is the 4-vector angular momentum of molecule i in its body frame, and $\hat{L}_{\hat{x},i}$, $\hat{L}_{\hat{y},i}$, and $\hat{L}_{\hat{z},i}$ are the components of the $\hat{\mathbf{L}}_i$ vector. For simplicity, the 4-vectors $\mathbf{P}_{\mathbf{q},i}^{(4)}$ and $\mathbf{q}_i^{(4)}$ are denoted by $\mathbf{P}_{\mathbf{q},i}$ and \mathbf{q}_i , respectively, in the following steps.

10. The quaternion momentum of every molecule i is updated to a half time step:

$$\mathbf{P}_{\mathbf{q},i}(t + \frac{\Delta t}{2}) = \mathbf{P}_{\mathbf{q},i}(t) + \frac{\Delta t}{2} \hat{\mathbf{T}}_{\mathbf{q},i}(t) \quad (\text{S55})$$

11. The following steps in Eqs. (S56) to (S65) are repeated for m times (e.g., $m = 10^{26}$). A larger value of m increases the accuracy of the scheme at the expense of a larger computational cost²⁹. For each molecule i , the quaternions are updated to the full time step:

$$\mathbf{P}_{\mathbf{q},i} = \cos(\phi_3 \delta t / 2) \mathbf{P}_{\mathbf{q},i} + \sin(\phi_3 \delta t / 2) \mathbf{D}_3 \mathbf{P}_{\mathbf{q},i} \quad (\text{S56})$$

$$\mathbf{q}_i = \cos(\phi_3 \delta t / 2) \mathbf{q}_i + \sin(\phi_3 \delta t / 2) \mathbf{D}_3 \mathbf{q}_i \quad (\text{S57})$$

$$\mathbf{P}_{\mathbf{q},i} = \cos(\phi_2 \delta t / 2) \mathbf{P}_{\mathbf{q},i} + \sin(\phi_2 \delta t / 2) \mathbf{D}_2 \mathbf{P}_{\mathbf{q},i} \quad (\text{S58})$$

$$\mathbf{q}_i = \cos(\phi_2 \delta t / 2) \mathbf{q}_i + \sin(\phi_2 \delta t / 2) \mathbf{D}_2 \mathbf{q}_i \quad (\text{S59})$$

$$\mathbf{P}_{\mathbf{q},i} = \cos(\phi_1 \delta t) \mathbf{P}_{\mathbf{q},i} + \sin(\phi_1 \delta t) \mathbf{D}_1 \mathbf{P}_{\mathbf{q},i} \quad (\text{S60})$$

$$\mathbf{q}_i = \cos(\phi_1 \delta t) \mathbf{q}_i + \sin(\phi_1 \delta t) \mathbf{D}_1 \mathbf{q}_i \quad (\text{S61})$$

$$\mathbf{P}_{\mathbf{q},i} = \cos(\phi_2 \delta t / 2) \mathbf{P}_{\mathbf{q},i} + \sin(\phi_2 \delta t / 2) \mathbf{D}_2 \mathbf{P}_{\mathbf{q},i} \quad (\text{S62})$$

$$\mathbf{q}_i = \cos(\phi_2 \delta t / 2) \mathbf{q}_i + \sin(\phi_2 \delta t / 2) \mathbf{D}_2 \mathbf{q}_i \quad (\text{S63})$$

$$\mathbf{P}_{\mathbf{q},i} = \cos(\phi_3 \delta t / 2) \mathbf{P}_{\mathbf{q},i} + \sin(\phi_3 \delta t / 2) \mathbf{D}_3 \mathbf{P}_{\mathbf{q},i} \quad (\text{S64})$$

$$\mathbf{q}_i(t + \Delta t) = \cos(\phi_3 \delta t / 2) \mathbf{q}_i + \sin(\phi_3 \delta t / 2) \mathbf{D}_3 \mathbf{q}_i \quad (\text{S65})$$

where $\delta t = \Delta t / m$, $\phi_k = (\mathbf{P}_{\mathbf{q},i}^T \mathbf{D}_k \mathbf{q}_i) / (4 \hat{I}_{kk,i})$ (for $k = 1, 2, 3$), in which $\hat{I}_{11,i} = \hat{I}_{\hat{x}\hat{x},i}$, $\hat{I}_{22,i} = \hat{I}_{\hat{y}\hat{y},i}$, and $\hat{I}_{33,i} = \hat{I}_{\hat{z}\hat{z},i}$, and:

$$\mathbf{D}_1 \mathbf{q}_i = (-q_{1,i} \quad q_{0,i} \quad q_{3,i} \quad -q_{2,i})^T \quad (\text{S66})$$

$$\mathbf{D}_2 \mathbf{q}_i = (-q_{2,i} \quad -q_{3,i} \quad q_{0,i} \quad q_{1,i})^T \quad (\text{S67})$$

$$\mathbf{D}_3 \mathbf{q}_i = (-q_{3,i} \quad q_{2,i} \quad -q_{1,i} \quad q_{0,i})^T \quad (\text{S68})$$

$\mathbf{D}_k \mathbf{P}_{\mathbf{q},i}$ ($k = 1, 2, 3$) are similarly computed, where the quaternion components are replaced by the corresponding components of the quaternion momenta:

$$\mathbf{D}_1 \mathbf{P}_{\mathbf{q},i} = (-P_{q,1,i} \quad P_{q,0,i} \quad P_{q,3,i} \quad -P_{q,2,i})^T \quad (\text{S69})$$

$$\mathbf{D}_2 \mathbf{P}_{\mathbf{q},i} = (-P_{q,2,i} \quad -P_{q,3,i} \quad P_{q,0,i} \quad P_{q,1,i})^T \quad (\text{S70})$$

$$\mathbf{D}_3 \mathbf{P}_{\mathbf{q},i} = (-P_{q,3,i} \quad P_{q,2,i} \quad -P_{q,1,i} \quad P_{q,0,i})^T \quad (\text{S71})$$

12. The new rotation matrix \mathbf{R}_i is computed for each molecule i , using Eq. (S47), based on the new quaternions obtained after applying Eqs. (S56) to (S65).

13. The new positions of atoms are computed using the new rotation matrix:

$$\mathbf{r}_{ji} = \mathbf{R}_i \hat{\mathbf{r}}_{ji} \quad (\text{S72})$$

where \mathbf{r}_{ji} and $\hat{\mathbf{r}}_{ji}$ are the position vectors of atom j in molecule i , with respect to the center-of-mass of i , in the laboratory and body frames ($\hat{\mathbf{r}}_{ji}$ is computed in step 4), respectively. Using \mathbf{r}_{ji} and the center-of-mass coordinates, the positions of the atoms are updated.

14. The angular momentum and angular velocity of every molecule i are updated to a half time step:

$$\hat{\mathbf{L}}_i^{(4)} = \frac{1}{2} \mathbf{M}_i^{-1} \mathbf{P}_{\mathbf{q},i} \quad (\text{S73})$$

$$\hat{\omega}_{k,i} = \hat{L}_{k,i} / \hat{I}_{kk,i} \quad (k = 1, 2, 3)$$

in which $k = 1$, $k = 2$, and $k = 3$ correspond to the \hat{x} , \hat{y} , and \hat{z} axes, respectively.

15. The new torques in laboratory frame and body frame, as well as the new quaternion torques are computed at half time step, as demonstrated in steps 5-7.

16. The quaternion momentum of each molecule i is updated from a half time step (output of step 11) to a full time step:

$$\mathbf{P}_{\mathbf{q},i}(t + \Delta t) = \mathbf{P}_{\mathbf{q},i}(t + \frac{\Delta t}{2}) + \frac{\Delta t}{2} \hat{\mathbf{T}}_{\mathbf{q},i}(t + \frac{\Delta t}{2}) \quad (\text{S74})$$

17. The new angular momenta and angular velocities are computed for all molecules at a full time step, similar to step 14.

18. For an MD trajectory of length N_{step} , steps 10-17 are repeated for $N_{\text{step}} - 1$ times.

19. The new rotational kinetic energy of the system, $K_{\text{new}}^{\text{rot}}$, is computed based on the final angular velocities:

$$K_{\text{new}}^{\text{rot}} = \sum_{i=1}^N \frac{1}{2} (\hat{I}_{\hat{x}\hat{x},i} \hat{\omega}_{\hat{x},i}^2 + \hat{I}_{\hat{y}\hat{y},i} \hat{\omega}_{\hat{y},i}^2 + \hat{I}_{\hat{z}\hat{z},i} \hat{\omega}_{\hat{z},i}^2) \quad (\text{S75})$$

where i is the molecule number and N is the total number of molecules in the system.

20. The trial move is accepted or rejected according to the acceptance rule³:

$$\text{acc}(o \rightarrow n) = \min(1, \exp[-\beta(\Delta U + \Delta K^{\text{rot}})]) \quad (\text{S76})$$

where o and n denote the old and new (initial and final) configurations on the MD trajectory, and ΔU and ΔK^{rot} are the differences in potential energy and rotational kinetic energy, respectively, between the old and new configurations. β is defined as $1/(k_{\text{B}}T)$, where k_{B} is the Boltzmann constant, and T is the absolute temperature. For more details on this rigid body dynamics integrator, the reader is referred to Refs.²⁷ and²⁹.

Simulation Details for the Case Studies

Thermodynamic Integration

All simulations were performed using Brick-CFCMC⁷. For the computation of $\mu_{\text{NaCl}}^{\text{ex}}$, the SPC/E³⁰ and Joung-Cheatham³¹ force fields were used for water and sodium chloride, respectively. The force field parameters used in the simulations are listed in Table S1. 101 independent MC simulations with different and fixed values of λ were performed at 298 K and 1 bar, in the *NPT* ensemble. Initial configurations for these simulations were generated with a box length of 20.8 Å, using 300 water molecules and a fractional group consisting of one sodium ion and one chloride ion. Atomic overlaps, caused by the random generation of initial configurations were removed by 10^3 initialization MC cycles, where only translation and rotation trial moves were used. Equilibration and production stages of the simulations were carried out, each for 10^6 MC cycles. In Brick-CFCMC, a single MC cycle consists of N MC trial moves where N is the number of molecules in the simulation box. During the equilibration and production cycles, different trial moves were performed with fixed probabilities: translations (49.49%), hybrid translations (0.01%), rotations (49.5%), and volume changes (1%). LJ interactions were truncated at 10 Å. Analytic tail corrections²¹ were applied, and the Lorentz-Berthelot mixing rules²¹ were used to compute interaction parameters for different atom types. For electrostatic interactions, the damped and shifted version of the Wolf method was used¹². The Wolf method parameters were set to 8.25 Å and 0.22 \AA^{-1} for the cutoff radius and damping parameter, respectively. For the simulations using the Ewald summation for electrostatic interactions, a relative precision of 10^{-6} was used. In post-processing, $\langle \frac{\partial U}{\partial \lambda} \rangle$ was integrated using a tool provided with Brick-CFCMC. This tool fits a spline to $\langle \frac{\partial U}{\partial \lambda} \rangle$, and subsequently integrates the spline from $\lambda = 0$ to $\lambda = 1$ using the trapezoidal rule.

Table S1: Force field parameters used in the MC simulations of NaCl/water solutions. For water and NaCl molecules, the SPC/E³⁰ and Joung-Cheatham³¹ force fields were used, respectively.

Atom	ϵ/k_B / [K]	σ / [Å]	q / [e^-]
O _{H₂O}	78.177	3.166	-0.8476
H _{H₂O}	1.0000	1.000	0.4238
Na _{NaCl}	177.46	2.159	1.0000
Cl _{NaCl}	6.4340	4.830	-1.0000

Hybrid Translation Trial Moves

The optimal time step size (Δt) for the hybrid translation trial move was obtained for a system of a choline chloride/urea (ChCIU) deep eutectic solvent (DES) at 338.15 K and 1 bar, in the *NPT* ensemble. The Generalized AMBER force field (GAFF)³² parameters were used for the DES, and the charges of cation and anion were scaled by 0.8³³. All force field parameters are tabulated in the Supporting Information of Ref.³⁴. 100 urea molecules and 50 choline chloride ion pairs were used in the simulations. A well-equilibrated configuration of ChCIU was used as initial configuration. Independent runs were performed with different values of the time step, and the average acceptance probabilities and displacements were computed. All runs consisted of 500 production MC cycles (no equilibration), during which only the hybrid translation trial move was carried out with a trajectory length of 5 timesteps. The Ewald summation method¹⁰, with $k = 8$ and a damping parameter of $\alpha = 0.3 \text{ \AA}^{-1}$ was used to compute long-range electrostatic energies. For the computation of electrostatic forces in the short MD trajectories, the damped, shifted Wolf method¹² was used, which is computationally less expensive than the Ewald summation. The damping parameter of the damped, shifted Wolf method was set to 0.2 \AA^{-1} . The cutoff radius was set to 10 \AA for all short-range energies and forces. Analytic tail corrections²¹ were used for the long-range LJ interactions, and the Lorentz-Berthelot mixing rules²¹ were applied to compute the LJ interactions between non-identical atom types.

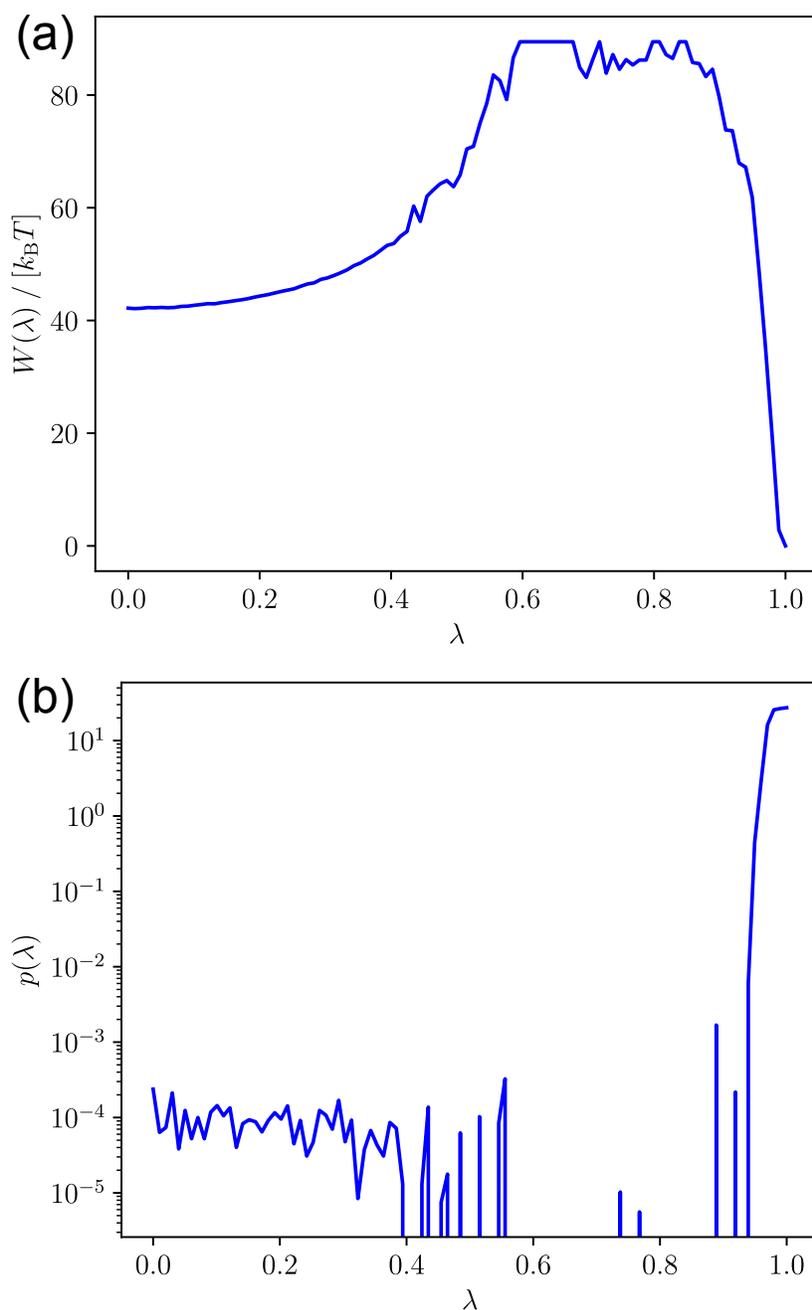


Figure S2: (a) The weight function and (b) probability distribution of the NaCl fractional group as a function of λ when making an attempt to compute the excess chemical potential of NaCl at infinite dilution in water from a single simulation. From subfigure (a), it can be seen that the biasing function reaches up to ca. $90 k_B T$. As a result, same interval with very large biasing function is not sampled at all. From subfigure (b), it can also be seen that only very high λ values are sampled. To obtain a flat probability distribution of λ , multiple simulations with confined λ -space will be required.

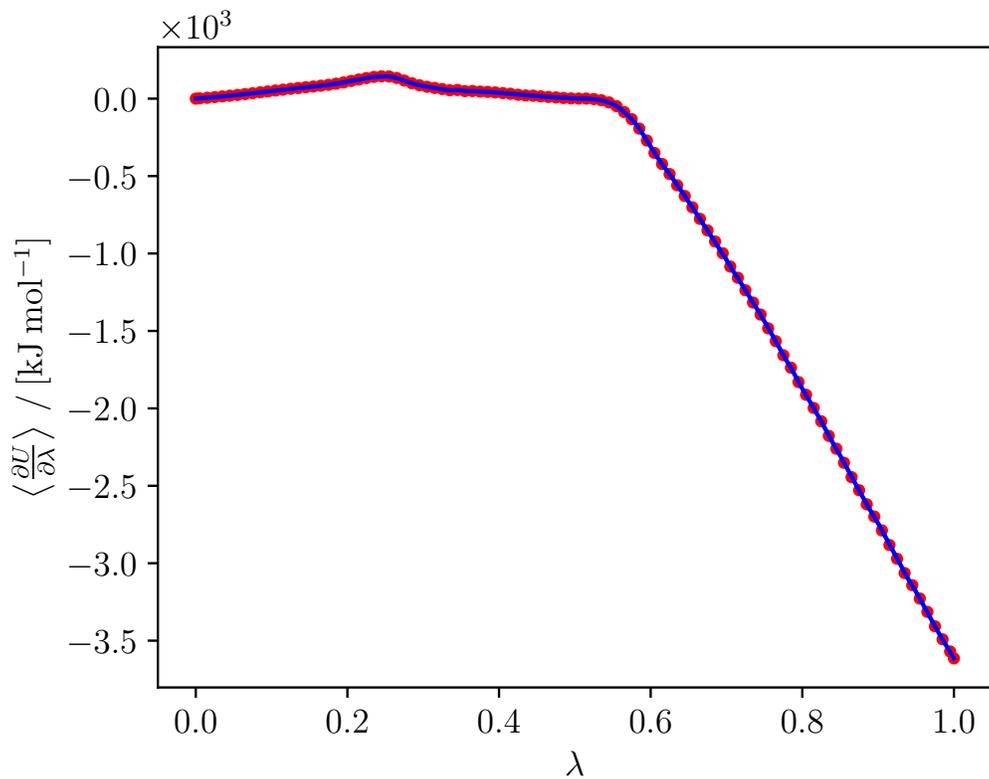


Figure S3: The value of $\langle \frac{\partial U}{\partial \lambda} \rangle$ as a function of λ for infinitely diluted NaCl in water at 298 K and 1 bar. In these simulations, Ewald summation with a relative precision of 10^{-6} was used for the electrostatic interactions. The red circles represent the fitted spline while the blue line represents the values of $\langle \frac{\partial U}{\partial \lambda} \rangle$.

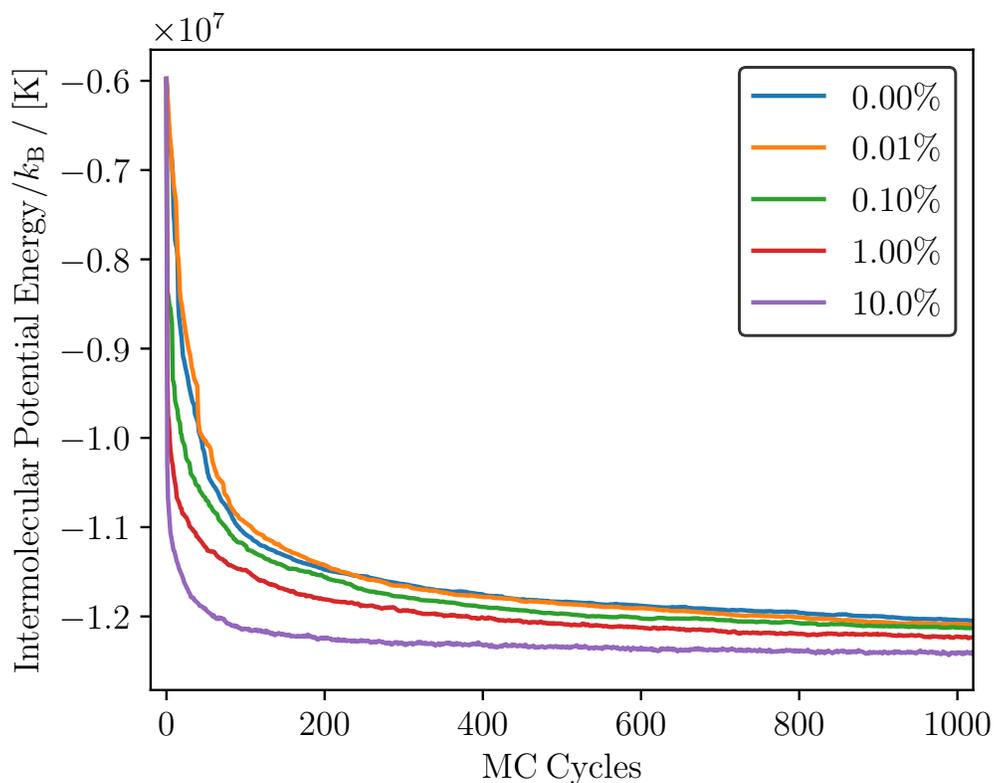


Figure S4: Intermolecular potential energy as a function of the number of MC cycles for different fractions of hybrid MD/MC trial moves. The other trial moves are single-molecule displacement and rotations. The numbers of attempted rotations and translation are equal, both for hybrid MD/MC trial moves and single-molecule trial moves. The simulations were performed for choline chloride/urea DES at 323 K in NVT ensemble. For both hybrid MD/MC translation and hybrid MD/MC rotation moves, a timestep of 1 fs and a trajectory length of $N_{\text{step}} = 10$ were used. The results show that the use of hybrid MD/MC moves significantly facilitates the equilibration of the simulation box.

References

- (1) Shi, W.; Maginn, E. J. Continuous Fractional Component Monte Carlo: An Adaptive Biasing Method for Open System Atomistic Simulations. *J. Chem. Theory Comput.* **2007**, *3*, 1451–1463.
- (2) Rahbari, A.; Hens, R.; Ramdin, M.; Moulτος, O.; Dubbeldam, D.; Vlugt, T. J. H. Recent advances in the continuous fractional component Monte Carlo methodology. *Mol. Simul.* **2021**, *47*, 804–823.
- (3) Frenkel, D.; Smit, B. *Understanding Molecular Simulation*, 2nd ed.; Elsevier: San Diego, California, 2002; Vol. 1.
- (4) Kirkwood, J. G. Statistical Mechanics of Fluid Mixtures. *The Journal of Chemical Physics* **2004**, *3*, 300.
- (5) Pham, T. T.; Shirts, M. R. Identifying low variance pathways for free energy calculations of molecular transformations in solution phase. *J. Chem. Phys.* **2011**, *135*, 034114.
- (6) Beutler, T. C.; Mark, A. E.; van Schaik, R. C.; Gerber, P. R.; van Gunsteren, W. F. Avoiding singularities and numerical instabilities in free energy calculations based on molecular simulations. *Chem. Phys. Lett.* **1994**, *222*, 529–539.
- (7) Hens, R.; Rahbari, A.; Caro-Ortiz, S.; Dawass, N.; Erdos, M.; Poursaeidesfahani, A.; Salehi, H. S.; Celebi, A. T.; Ramdin, M.; Moulτος, O. A.; Dubbeldam, D.; Vlugt, T. J. H. Brick-CFCMC: Open Source Software for Monte Carlo Simulations of Phase and Reaction Equilibria Using the Continuous Fractional Component Method. *J. Chem. Inf. Model.* **2020**, *60*, 2678–2682.
- (8) Lindahl, E.; Abraham, M.; Hess, B.; van der Spoel, D. GROMACS 2021.2 Manual. 2021; <https://doi.org/10.5281/zenodo.4723561>.

- (9) Abraham, M. J.; Murtola, T.; Schulz, R.; Páll, S.; Smith, J. C.; Hess, B.; Lindahl, E. GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX* **2015**, *1-2*, 19–25.
- (10) Ewald, P. Die Berechnung optischer und elektrostatischer Gitterpotentiale. *Ann. Phys.* **1921**, *369*, 253–287.
- (11) Wolf, D.; Keblinski, P.; Phillpot, S.; Eggebrecht, J. Exact method for the simulation of Coulombic systems by spherically truncated, pairwise r^{-1} summation. *J. Chem. Phys.* **1999**, *110*, 8254–8282.
- (12) Fennell, C. J.; Gezelter, J. D. Is the Ewald summation still necessary? Pairwise alternatives to the accepted standard for long-range electrostatics. *J. Chem. Phys.* **2006**, *124*, 234104.
- (13) Vlugt, T. J. H.; Garcia-Perez, E.; Dubbeldam, D.; Ban, S.; Calero, S. Computing the Heat of Adsorption using Molecular Simulations: The Effect of Strong Coulombic Interactions. *J. Chem. Theory Comput.* **2008**, *4*, 1107–1118.
- (14) Wells, B. A.; Chaffee, A. L. Ewald Summation for Molecular Simulations. *J. Chem. Theory Comput.* **2015**, *11*, 3684–3695.
- (15) Hens, R.; Vlugt, T. J. H. Molecular Simulation of Vapor–Liquid Equilibria Using the Wolf Method for Electrostatic Interactions. *J. Chem. Eng. Data* **2017**, *63*, 1096–1102.
- (16) Waibel, C.; Feinler, M. S.; Gross, J. A Modified Shifted Force Approach to the Wolf Summation. *J. Chem. Theory Comput.* **2018**, *15*, 572–583.
- (17) Zahn, D.; Schilling, B.; Kast, S. M. Enhancement of the Wolf Damped Coulomb Potential: Static, Dynamic, and Dielectric Properties of Liquid Water from Molecular Simulation. *J. Phys. Chem. B* **2002**, *106*, 10725–10732.

- (18) Demontis, P.; Spanu, S.; Suffritti, G. B. Application of the Wolf method for the evaluation of Coulombic interactions to complex condensed matter systems: Aluminosilicates and water. *J. Chem. Phys.* **2001**, *114*, 7980–7988.
- (19) Rahbari, A.; Hens, R.; Jamali, S.; Ramdin, M.; Dubbeldam, D.; Vlugt, T. J. H. Effect of truncating electrostatic interactions on predicting thermodynamic properties of water–methanol systems. *Mol. Simul.* **2018**, *45*, 336–350.
- (20) Rossky, P.; Doll, J.; Friedman, H. Brownian dynamics as smart Monte Carlo simulation. *J. Chem. Phys.* **1978**, *69*, 4628–4633.
- (21) Allen, M. P.; Tildesley, D. J. *Computer Simulation of Liquids*, 2nd ed.; Oxford University Press: New York, NY, USA, 2017.
- (22) Moučka, F.; Rouha, M.; Nezbeda, I. Efficient multiparticle sampling in Monte Carlo simulations on fluids: Application to polarizable models. *J. Chem. Phys.* **2007**, *126*, 224106.
- (23) Verlet, L. Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules. *Phys. Rev.* **1967**, *159*, 98–103.
- (24) Swope, W. C.; Andersen, H. C.; Berens, P. H.; Wilson, K. R. A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters. *J. Chem. Phys.* **1982**, *76*, 637–649.
- (25) Duane, S.; Kennedy, A.; Pendleton, B. J.; Roweth, D. Hybrid Monte Carlo. *Phys. Lett. B* **1987**, *195*, 216–222.
- (26) Smith, W.; Yong, C.; Rodger, P. DL_POLY: Application to molecular simulation. *Mol. Simul.* **2002**, *28*, 385–471.

- (27) Miller, T.; Eleftheriou, M.; Pattnaik, P.; Ndirango, A.; Newns, D.; Martyna, G. Symplectic quaternion scheme for biophysical molecular dynamics. *J. Chem. Phys.* **2002**, *116*, 8649–8659.
- (28) Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; Flannery, B. P. *Numerical Recipes: The Art of Scientific Computing*, 3rd ed.; Cambridge University Press: New York, NY, USA, 2007.
- (29) Kamberaj, H.; Low, R.; Neal, M. Time reversible and symplectic integrators for molecular dynamics simulations of rigid molecules. *J. Chem. Phys.* **2005**, *122*, 224114.
- (30) Berendsen, H.; Grigera, J.; Straatsma, T. The missing term in effective pair potentials. *J. Phys. Chem.* **1987**, *91*, 6269–6271.
- (31) Joung, I. S.; Cheatham, T. E. Determination of Alkali and Halide Monovalent Ion Parameters for Use in Explicitly Solvated Biomolecular Simulations. *J. Phys. Chem. B* **2008**, *112*, 9020–9041.
- (32) Wang, J.; Wolf, R. M.; Caldwell, J. W.; Kollman, P. A.; Case, D. A. Development and testing of a general amber force field. *J. Comput. Chem.* **2004**, *25*, 1157–1174.
- (33) Perkins, S. L.; Painter, P.; Colina, C. M. Molecular Dynamic Simulations and Vibrational Analysis of an Ionic Liquid Analogue. *J. Phys. Chem. B* **2013**, *117*, 10250–10260.
- (34) Salehi, H. S.; Hens, R.; Moulton, O. A.; Vlugt, T. J. H. Computation of gas solubilities in choline chloride urea and choline chloride ethylene glycol deep eutectic solvents using Monte Carlo simulations. *J. Mol. Liq.* **2020**, *316*, 113729.