

Knowledge Distillation of Noisy Force Labels for Improved Coarse-Grained Force Fields

Feranmi V. Olowookere^{1,4}, Sakib Matin², Aleksandra Pachaliev³, Nicholas Lubbers¹, and Emily Shinkle¹

¹*Computing and Artificial Intelligence Division, Los Alamos National Laboratory, Los Alamos, NM 87545, USA*

²*Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87545, USA*

³*Earth and Environmental Sciences Division, Los Alamos National Laboratory, Los Alamos, NM 87545, USA*

⁴*The University of Alabama, Tuscaloosa, AL 35487, USA*

Abstract

Molecular dynamics simulations are an integral tool for studying the atomistic behavior of materials under diverse conditions. However, they can be computationally demanding in wall-clock time, especially for large systems, which limits the time and length scales accessible. Coarse-grained (CG) models reduce computational expense by grouping atoms into simplified representations commonly termed *beads*, but sacrifice atomic detail and introduce mapping noise, complicating the training of machine-learned surrogates. Moreover, because CG models inherently include entropic contributions, they cannot be fit directly to all-atom energies, leaving instantaneous, noisy forces as the only state-specific quantities available for training. Here, we apply a knowledge distillation framework by first training an initial CG neural network potential (the *teacher*) solely on CG-mapped forces to denoise those labels, then distill its force and energy predictions to train refined CG models (the *student*) in both single- and ensemble-training setups while exploring different force and energy target combinations. We validate this framework on a complex molecular fluid—a deep eutectic solvent—by evaluating two-, three-, and many-body properties and compare the CG and all-atom results. Our findings demonstrate that training a student model on ensemble teacher-predicted forces and per-bead energies improves the quality and stability of CG force fields.

1 Introduction

All-atom (AA) molecular dynamics (MD) simulations are widely used to probe structure, thermodynamics, and transport,^{1–4} but generating long trajectories or large ensembles from these simulations could be computationally demanding in practice, especially for larger systems.⁵ Coarse-grained (CG) models mitigate this cost by sacrificing fidelity, mapping groups of atoms into units (often called *beads*), thereby reducing both the number of particles and the number of interactions that must be computed.^{6,7} Additionally, the resulting variables typically evolve on a smoother potential energy surface (PES) than the underlying AA system,^{8,9} which in turn accelerates the exploration and sampling of the system’s energy states.

CG force fields (FFs) are typically derived via either top-down or bottom-up protocols.^{6,9} The former approach chooses bead types and interaction forms to reproduce experimental observables or thermodynamic targets,^{10,11} but they may sacrifice structural fidelity; even widely used MARTINI-based models^{11,12} can underpredict or misrepresent particle coordination when a single bead choice must best represent diverse local chemistries.^{13,14} On the other hand, the bottom-up approach derives effective interactions directly from AA data by matching forces or distribution functions.^{15–18} Yet, they are state-dependent and struggle to capture many-body effects arising from the averaged-out AA degrees of freedom.^{9,18,19} More recently, data-driven CG models have demonstrated that learning flexible, many-body bead interactions can improve accuracy and transferability compared to pairwise CG approaches.^{20–23} Although, faithfully reproducing dynamical properties still remains an open challenge; we refer the reader to recent efforts in improving CG dynamics.^{24–28}

Machine learning (ML) methods for CG have lately advanced alongside ML-based development of AA force fields.^{29–31} In particular, architectures originally designed for AA potentials³² naturally extend from atom-wise contributions to the system energy into bead-wise contributions to the CG free energy. It is important to note that a bottom-up CG model is bound to

the underlying accuracy of its AA FF, so any ML workflow should yield consistent performance regardless of which AA reference is used.

Bottom-up CG training includes two important and related challenges. First, projecting AA forces $\mathbf{f}(\mathbf{r})$ from AA positions $\mathbf{r}$ onto bead forces $\mathbf{F}(\mathbf{R})$ over bead positions $\mathbf{R}$ smooths out detailed fluctuations, but data produced directly from AA MD still contains the unsmoothed, noisy labels.^{9,33} Hence, one must sample enough CG configurations ($> 10^6$ data points total) to ensure that the error is dominated by variations in the CG PES rather than by noise in the force labels.³⁴ Second, the effective CG energy functions are potentials of mean force (PMF), a form of *free energy*, and therefore include entropic contributions that are not tractable to compute. Although one can include AA energies in a CG loss function, doing so yields poor CG results, so in practice only the instantaneous, noisy force labels are used for training the model.¹⁸

To address these limitations, knowledge distillation (KD)³⁵ provides a training paradigm in which an initial *teacher* model guides a more accurate *student* model by supplying additional supervision signals.^{35–37} In other words, combining noisy labels with the *teacher’s* denoised outputs enhances the accuracy of the *student*. Originally applied to image classification,³⁵ KD has recently been adapted in molecular simulations to enhance the accuracy and efficiency of ML interatomic potentials.^{38–42} Distillation from an *ensemble* of teachers further reduces variance and yields smoother, more reliable targets than any single model,^{43–45} which is valuable especially when labels are noisy. To our knowledge, KD has not been applied to the CG domain, where supervision is dominated by noisy projected forces and where intermediate energy signals could be particularly beneficial; building CG models is highly related to the task of de-noising,⁴⁶ and models of intractable free energies can be built using derivative information alone.⁴⁷

Here, we introduce a KD-based workflow for ML CG potentials using the Hierarchically Interacting Particle Neural Network with Tensor Sensitivity (HIP-NN-TS) architecture,^{48,49} recently shown to construct thermodynamically transferable CG models.²⁰ The crucial obser-

vation behind our work is that initial teacher models perform estimation both of the denoised forces and forms an implicit integration for the intractable free energy, and that these quantities can be used as additional supervision signals for second-generation models to mitigate the difficulties associated with bottom-up CG training. We validate our framework on a complex molecular fluid, namely, a deep eutectic solvent (DES), by first training the teacher only on CG-mapped forces extracted from AA simulations. We then train student models on those CG-mapped forces along with various combinations of teacher-generated targets—forces, per-bead energies, and system energy. We show that students distilled from an ensemble of teachers and supervised with teacher forces along with per-bead energies achieve better accuracy of structural properties while maintaining single-model inference speed.

2 Method

Our simulation workflow for training and validating teacher and student models is outlined in Figure 1. We begin by generating AA data with MD simulations and mapping this data to a CG representation. Next, we train the teacher networks on this mapped AA data, and subsequently train student networks on the same reference set augmented with the additional supervision signals (forces and energies) generated by the teachers.

Generation of AA data for CG training. We generated AA data by simulating a system of 1,000 DES molecules composed of 250 choline, 250 chloride, and 500 urea molecules using the GROMACS package.⁵⁰ Bonded and non-bonded interactions were modeled with the Optimized Potentials for Liquid Simulations - All-Atom (OPLS-AA) force field parameterized for DESs.^{51,52} The initial configuration was built with Packmol,⁵³ using a cubic box of 44.84 Å that corresponds to the experimentally determined density of 1.196 g cm⁻³.⁵⁴ This was followed by energy minimization using the steepest-descent algorithm. We then equilibrated for 5 ns in the canonical (NVT) ensemble at 298.1 K with a Nosé-Hoover thermostat (0.5 ps time constant).⁵⁵

Following equilibration, we performed a 1 ns production run in the NVT ensemble and saved snapshots every 1 ps, resulting in 1,000 configurations. Throughout equilibration and production, we used a 2 fs time step with periodic boundary conditions applied in all directions; hydrogen bond lengths were constrained with Linear Constraint Solver (LINCS) algorithm.⁵⁶ Dispersive and electrostatic interactions were truncated at 10 Å and long-range electrostatics were computed via the Particle Mesh Ewald method.⁵⁷

The AA trajectories were mapped to the CG representation by applying our bead-mapping protocol (discussed below) to AA positions and forces, producing the dataset used to train ML CG models.

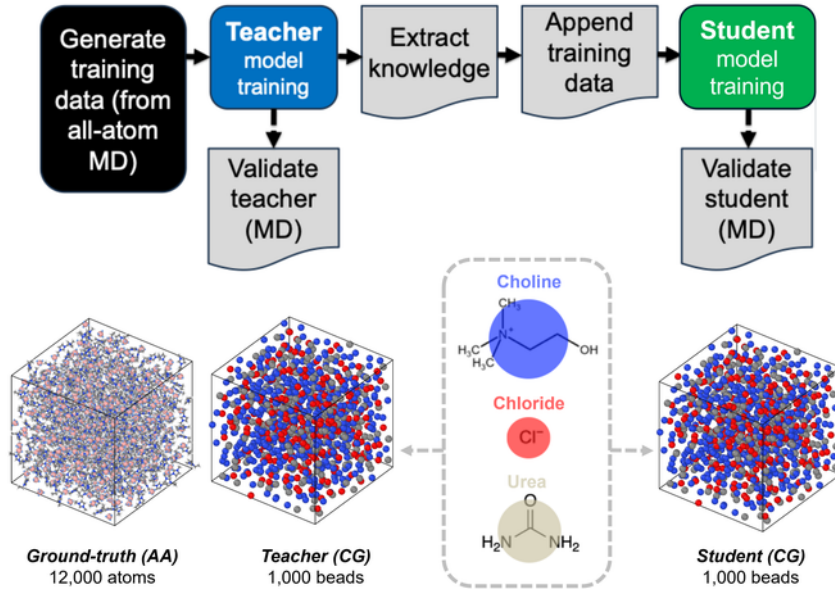


Figure 1: Simulation workflow for training and validating teacher and student models. MD: molecular dynamics, AA: all-atom, CG: coarse-grained. Each molecule is represented by one bead at the coarse-grained level.

Coarse-Graining Scheme. Following our previous work,²⁰ we use a bottom-up strategy to build the CG force field.

$$\langle -\nabla U_{AA} \rangle = -\nabla U_{CG}. \quad (1)$$

As shown in Eq. 1, we define the CG free-energy surface, or PMF, so that it matches the

AA PES over a representative range of configurations in phase space, and we require the CG forces derived from this surface to reproduce the AA force averages conditioned on each CG configuration. In this study, each molecule is represented by a single bead; the DES therefore contains three bead types: choline (Cho), chloride (Cl), and urea, as shown in Figure 1.

CG positions are computed as centers of mass of the atoms assigned to each bead, and each bead mass is the sum of the atomic masses in the corresponding molecule. The force on a CG bead is taken as the sum of the atomic forces within that bead, which provides an unbiased estimator of the negative gradient of the CG free energy with respect to the bead coordinates. Although alternative mapping schemes exist,⁵⁸ the force-mapping operator must remain consistent with the chosen coordinate mapping; specifically, after contraction over atomistic indices, it should act as the inverse of the AA Jacobian of the coordinate map. Throughout this study, we treat the CG-mapped data obtained from the AA trajectory as the “ground-truth” for training.

Model architecture. For both teacher and student CG models, we use the HIP-NN-TS,^{48,49} a graph convolutional neural network that represents the system energy as a sum of per-bead contributions computed through a hierarchy of interaction and bead-environment layers. It extends the original HIP-NN model⁴⁸ by incorporating a tensor sensitivity component, enabling each neuron in the interaction layer to encode many-body features. The convolutions operate on pairwise displacement vectors and thus naturally produce invariance to rotations, translations, and permutations of identical beads. The network featurizes each bead’s local environment using its type and neighbor displacement vectors, then predicts energy contributions at each hierarchy which are summed to yield per-bead energies ε and the system energy E . Forces on each bead are then obtained by automatic differentiation of E with respect to the bead positions.

The HIP-NN-TS hyperparameters in the ML CG models are shown in Table 1, selected by trial and error in our previous work²⁰ and have performed well in prior HIP-NN-TS applications

Table 1: HIP-NN-TS hyperparameters.

Parameter	Symbol	Value
Tensor order	ℓ	2
Interaction layers	n_{int}	1
Atomic-environment layers	n_{atom}	3
Sensitivity functions	n_v	20
Atomic features per layer	n_{feature}	32

to atomistic simulation.^{38,48} To ensure stability in sparsely sampled regions, we supplement the learned potential with a physics-based short-range repulsive term. This additional pairwise potential is activated whenever two beads approach closer than a cutoff r_0 , defined by the radial distribution function (RDF), thereby preventing unphysical overlaps and avoiding extrapolation of the ML component into untrained, small- r regimes.

$$E_{\text{rep}}(r) = E_0 e^{-ar}, \quad (2)$$

The repulsive term takes the form in Eq. 2, where r is the inter-bead distance and $E_0, a > 0$ are system-specific parameters.²⁰ Example training scripts for both teacher and student models are available in the open-source `hippynn` repository.⁵⁹

Teacher–Student training overview. We train one or more teacher models on the noisy ground-truth forces $\mathbf{F}_i$ and then use the trained model(s) to generate additional supervision signals. These signals are appended to the initial training set to guide the learning of an improved (more accurate and/or faster) student model. The teacher-provided supervision captures richer structure in the data; in this study, it includes per-bead energies ε , total molecular energy E , and forces $\mathbf{f}_i$. We train the student in two settings: using the knowledge from a single teacher or from an ensemble of eight teachers. Although ensemble size can be varied, we used eight for all experiments; preliminary tests with four teachers produced similar results, thus eight was chosen to ensure robust averaging throughout this study.

Teacher model training. Eight teacher ($\mathbf{T}$) models were trained on the same dataset using

an identical architecture and size, differing only by their random initialization seeds. The teachers are trained on the AA-to-CG mapped dataset $\mathcal{D} : \{(\mathbf{R}_i, Z_i)\} \mapsto \{\mathbf{F}_i\}$, where $\mathbf{R}_i$ are i th bead positions, Z_i are bead types, and $\mathbf{F}_i$ are bead forces. Training uses stochastic gradient descent to minimize the loss function:

$$\mathcal{L}_{\text{teacher}} = w_F \mathcal{L}_{\text{err}}(\hat{\mathbf{F}}, \mathbf{F}). \quad (3)$$

The error term in Eq. 3 is defined as an equal-weight sum of root mean squared error (RMSE) and mean absolute error (MAE), which has proved successful for training previous HIP-NN-TS models^{20,38,43}

$$\mathcal{L}_{\text{err}}(\hat{y}, y) = \text{RMSE}(\hat{y}, y) + \text{MAE}(\hat{y}, y), \quad (4)$$

where $\hat{y}$ denotes the model prediction and y the target in Eq. 4. In addition to using the eight independently trained teachers ($\mathbf{T}$) during MD inference, we also deploy their averaged ensemble, denoted as $\mathbf{T8}$. After training, we extract ε , E , and f , to construct an augmented dataset for student model training.

Variants of knowledge distillation. We train the student in two regimes: **S1** (supervision from a single teacher $\mathbf{T}$) and **S8** (ensemble-averaged supervision from eight teachers $\mathbf{T8}$). Student variants are defined by their supervision signals in Table 2, varying the force target (ground-truth F only, teacher f only, or both) and the energy target (atomic energy ε only, system energy E only, or both). Student model names follow the convention of the training regime (**S1** or **S8**), followed by the included targets. For example, **S1F ε** denotes a student trained with single-teacher guidance using losses on ground-truth forces and teacher-provided per-bead energies only. As another example, **S8FfE ε** denotes a student trained with ensemble guidance using losses on all the targets.

Table 2 lists only a subset of all target combinations because we restricted the experiment matrix to the questions of greatest interest. Beginning with ε as the baseline, we toggled

Table 2: Variants of Knowledge Distillation for Student Models

Variant	Teacher type	System energy $\mathbf{E}$	Per-bead energy ϵ	Denoised forces $\mathbf{f}$	AA forces $\mathbf{F}$
$\mathbf{S1F}\epsilon$	Single	✗	✓	✗	✓
$\mathbf{S1f}\epsilon$	Single	✗	✓	✓	✗
$\mathbf{S1Ff}\epsilon$	Single	✗	✓	✓	✓
$\mathbf{S8Ff}$	Ensemble	✗	✗	✓	✓
$\mathbf{S8F}\epsilon$	Ensemble	✗	✓	✗	✓
$\mathbf{S8f}\epsilon$	Ensemble	✗	✓	✓	✗
$\mathbf{S8Ff}\epsilon$	Ensemble	✗	✓	✓	✓
$\mathbf{S8FfE}$	Ensemble	✓	✗	✓	✓
$\mathbf{S8FfE}\epsilon$	Ensemble	✓	✓	✓	✓

the force inputs $\mathbf{F}_i$ and $\mathbf{f}_i$, to identify the best force configuration and teacher type; we then fixed those choices and varied the energy targets ($\mathbf{E}$ and/or ϵ), in order to probe the most informative cases while keeping the number of training runs manageable.

Student model training. Depending on the target, each student variant is trained on the augmented dataset $\mathcal{D} : \{(\mathbf{R}_i, \mathbf{Z}_i)\} \mapsto \{\mathbf{F}_i, \mathbf{f}_i, \mathbf{E}, \epsilon_i\}$ (where i represents bead index) with the loss function:

$$\mathcal{L}_{\text{student}} = \mathcal{L}_{\text{teacher}} + w_f \mathcal{L}_{\text{err}}(\mathbf{f}_i^S, \mathbf{f}_i^T) + w_E \mathcal{L}_{\text{err}}(\mathbf{E}^S, \mathbf{E}^T) + w_\epsilon \mathcal{L}_{\text{err}}(\epsilon_i^S, \epsilon_i^T). \quad (5)$$

The alignment terms $\mathcal{L}_{\text{err}}(\mathbf{f}_i^S, \mathbf{f}_i^T)$, $\mathcal{L}_{\text{err}}(\mathbf{E}^S, \mathbf{E}^T)$, and $\mathcal{L}_{\text{err}}(\epsilon_i^S, \epsilon_i^T)$ in Eq. 5 encourage the student’s forces $\mathbf{f}_i^S$, per-bead energy partition ϵ_i^S , and system energy $\mathbf{E}^S$ to match those of the teacher’s $\mathbf{f}_i^T$, ϵ_i^T , and $\mathbf{E}^T$, respectively. It is important to note that we normalized the system-energy loss by the number of beads to prevent it from dominating the loss function relative to the other loss terms. To ensure a controlled comparison, both teacher and student models have the same architecture and size. All other settings and hyperparameters in Table 1 remained fixed, and the students differed only by the addition of the alignment term during training.

Table 3: Loss weights used for different target types in student models.

Target type used	System energy weight w_E	Per-bead energy weight w_ϵ	Denoised forces weight w_f	AA forces weight w_F
ϵ only	0	5	0	1
E only	5	0	0	1
Both ϵ and E	5	5	0	1
Both F and f	0	0	1	2

In Table 3, when only per-bead energies were used, the energy-loss weights were set to $w_\epsilon = 5$ and $w_E = 0$. When only the system energy was used, they were set to $w_\epsilon = 0$ and $w_E = 5$. When both per-bead and system energies were included, both energy-loss weights were set to $w_\epsilon = w_E = 5$. In variants that include both F and f , the force-loss term was partitioned as $w_F = 2$, $w_f = 1$, thereby emphasizing true forces while still leveraging the teacher’s denoising signal. Preliminary tests on these weights suggest that the chosen values yield stable optimization and faster convergence during training. We also found that scaling by factors of 2–5 produced similar validation trends, whereas larger imbalances caused the energy terms to overwhelm force learning despite normalization.

Model Validation and Analyses. We use MD to validate model performance, averaging results from eight independent replicas per model, with error bars denoting one standard deviation. Each replica was initialized with a configuration drawn from the training dataset and unique velocity seed, equilibrated for 1 ns, then sampled every 5 ps to yield 100 frames for model validation via the TRAVIS package.⁶⁰

During MD with the eight-teacher ensemble (**T8**), we evaluate all eight networks at each time step and update the positions with their averaged force. On the other hand, the distilled student (**S8**) is trained on these ensemble-averaged forces and energies, so it needs only a single network evaluation per step during inference. Equilibration was assessed by inspecting the potential-energy time series (see Figure S1) with the pymbar steady-state heuristic.⁶¹ Both

T and **S1** models reached equilibrium by 400 ps, whereas the **T8** and **S8** variants equilibrated very rapidly (< 2 ps).

We validated each CG model by comparing its simulation output with statistics from the reference AA trajectories. Because the CG training data contain noisy forces and lack explicit energy labels, regression metrics such as MAE, RMSE, and coefficient of determination (R^2) are unreliable; consequently, we judge model quality primarily through structural distribution functions. Specifically, we computed three complementary metrics to probe two-body, three-body, and many-body interactions. Dynamics were not examined here because CG models inherently sample more rapidly than AA counterparts due to the smoothed PES. Although methods^{24–28} such as including frictional forces²⁶ or applying scaling relationships^{62,63} have been proposed, addressing dynamics falls outside this study’s scope.

Two-body structure was quantified using the RDF,

$$g(\mathbf{r}) = \frac{\langle \rho(\mathbf{r}) \rangle}{\rho}, \quad (6)$$

which measures the probability of finding a particle at distance $\mathbf{r}$ from a reference particle, where $\langle \rho(\mathbf{r}) \rangle$ is the ensemble-averaged local number density of particles in a spherical shell at distance $\mathbf{r}$, and $\rho = N/V$ is the bulk number density of the system with N particles in volume V . To follow the standard notation $g(\mathbf{r})$ in Eq. 6, it is worth noting that we use $\mathbf{r}$ here as the inter-bead distances for both AA and CG RDFs.

Three-body structure was quantified by the angle distribution function (ADF). For each triplet (i, j, k) with central atom j , we compute

$$\theta_{ijk} = \cos^{-1}(\hat{\mathbf{R}}_{ji} \cdot \hat{\mathbf{R}}_{jk}) \quad (7)$$

and accumulate

$$P(\theta) = \frac{1}{N_{\text{triplet}}} \left\langle \sum_{j=1}^N \sum_{i \neq j} \sum_{\substack{k \neq j \\ k \neq i}} \delta(\theta - \theta_{ijk}) \right\rangle,$$

subject to the neighbor-cutoff conditions $\mathbf{R}_{ij}, \mathbf{R}_{jk} < \mathbf{R}_{\text{max}}$ (6.0 Å, 6.5 Å, or 7.5 Å). These values were chosen because they span the location of the first-solvation-shell peak in the corresponding RDFs, ensuring that the ADF captures the primary local structure. In Eq. 7, N_{triplet} is the total number of (i, j, k) triplets satisfying those cutoffs, and $\langle \dots \rangle$ denotes an ensemble average.

To quantify many-body aggregation, we follow the hierarchical clustering procedure of Frömbgen *et al.*⁶⁴ where two beads i and j belong to the same cluster when their separation $\mathbf{R}_{ij} = \|\mathbf{R}_i - \mathbf{R}_j\|$ is below a chosen cutoff $\mathbf{R}_{\text{cut}}$ ($\mathbf{R}_{\text{cut}} \leq L_{\text{box}}/2$ for a cubic box). Starting at $\mathbf{R}_{\text{cut}} = 0$ (all beads are isolated), we increase the cutoff continuously. Each time the criterion $\mathbf{R}_{ij} = \mathbf{R}_{\text{cut}}$ is met for the first time, two existing clusters merge; the corresponding cutoff value is stored as a merge distance $\mathbf{d}_m$. Plotting the histogram of all merge distances,

$$C(\mathbf{R}_{\text{cut}}) = \sum_m \delta(\mathbf{R}_{\text{cut}} - \mathbf{d}_m), \quad (8)$$

gives the cluster distribution function (CDF) in Eq. 8: a profile of how frequently clusters form as $\mathbf{R}_{\text{cut}}$ increases, reflecting the structural heterogeneity of the system.

To quantify deviations between the CG distributions $\mathbf{t}_1(\mathbf{R})$ and the ground-truth AA distributions $\mathbf{t}_2(\mathbf{R})$ (whether RDFs, ADFs, or CDFs), we employ a finite-sum approximation of the total absolute error (TAE)²⁰:

$$\text{TAE}(\mathbf{t}_1, \mathbf{t}_2) \approx \sum_{i=0}^n |\mathbf{t}_2(\mathbf{R}_i) - \mathbf{t}_1(\mathbf{R}_i)| \Delta \mathbf{R}, \quad (9)$$

In Eq. 9, $\{\mathbf{R}_i\}_{i=0}^n$ is a uniform grid of distances from $\mathbf{R}_0 = 0$ to $\mathbf{R}_n = \mathbf{R}_{\text{max}}$, $\Delta \mathbf{R} = \mathbf{R}_{i+1} - \mathbf{R}_i$ is the bin width, $\mathbf{t}_1$ and $\mathbf{t}_2$ denote the two probability densities (either $\mathbf{g}(\mathbf{R})$,

$P(\theta)$, or $C(R_{\text{cut}})$.

3 Results and Discussion

In this section, we report training metrics for our ML CG models and highlight the challenge of training teacher models solely on the initial dataset that contains only noisy forces. We then examine how single-teacher supervision compares with an ensemble of teachers when training student models, and we evaluate performance in both training and inference. Finally, we study the effect of different force and energy targets to identify an effective protocol for training the student.

Figure 2 previews our main findings. Despite identical architectures, single-teacher models with different initial training seeds produced highly variable, unstable dynamics with clustering. However, distilling the average predictions (forces and per-bead energies) of an ensemble of teachers into a single student improves the stability and accuracy, while delivering $\sim 5\times$ faster inference than the teacher ensemble. We quantify these trends below in training and inference metrics.

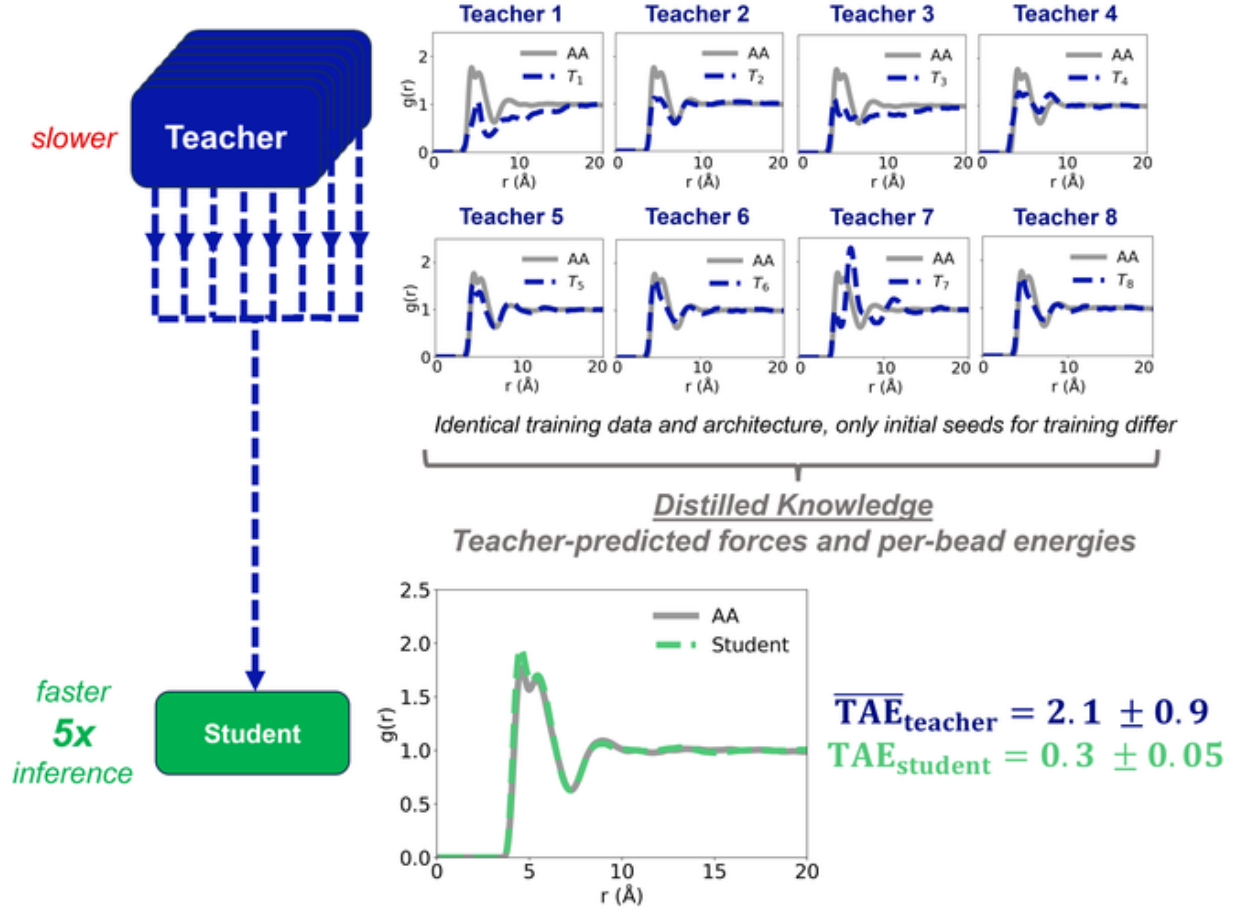


Figure 2: Distilling knowledge from an ensemble of teacher models into a single student improves both accuracy and efficiency of ML CG models. Each teacher is trained on the same AA force data but with different random seeds; averaging their predictions yields denoised forces and per-bead energies, which are then combined with the original training data to train the student. While the teachers exhibit bias in the RDF compared to the AA reference, the student both reproduces the reference RDF accurately and achieves roughly fivefold faster inference than the teacher-ensemble (**T8**) model.

3.1 Training Metrics

A common approach to evaluating ML models is to compare their predictions against the true values from held-out test data during training. For our teacher models, Figure 3 shows that across different random seeds, the MAE, RMSE, and coefficient of variation R^2 metrics exhibit little variation. Additionally, each teacher attains only $R^2 \approx 0.35$, which is low compared to typical ML benchmarks ($R^2 > 0.9$). This outcome is expected, since the instantaneous forces

$\mathbf{F}_i$ are drawn from a distribution which can be quite wide⁶⁵, whereas the network learns to estimate the conditional mean $\langle \mathbf{F}_i \rangle$, inherently adding noise to the loss.

Figure S3 presents the corresponding metrics for the student variant **S8**, distilled from the ensemble of eight teachers. Here, the force-prediction $\mathbf{R}^2$ changes only slightly ($0.35 < \mathbf{R}^2 < 0.38$) with the choice of energy targets. Likewise, the energy-prediction $\mathbf{R}^2$ for ε and $\mathbf{E}$ remains nearly the same ($\mathbf{R}^2 > 0.98$) except in cases where that quantity is omitted from training, in which case its $\mathbf{R}^2$ drops.

Overall, while MAE, RMSE, and $\mathbf{R}^2$ provide useful quantitative benchmarks, they alone are insufficient to assess ML CG model quality. Accordingly, we place greater emphasis on distributional and structural comparisons in the analyses that follow.

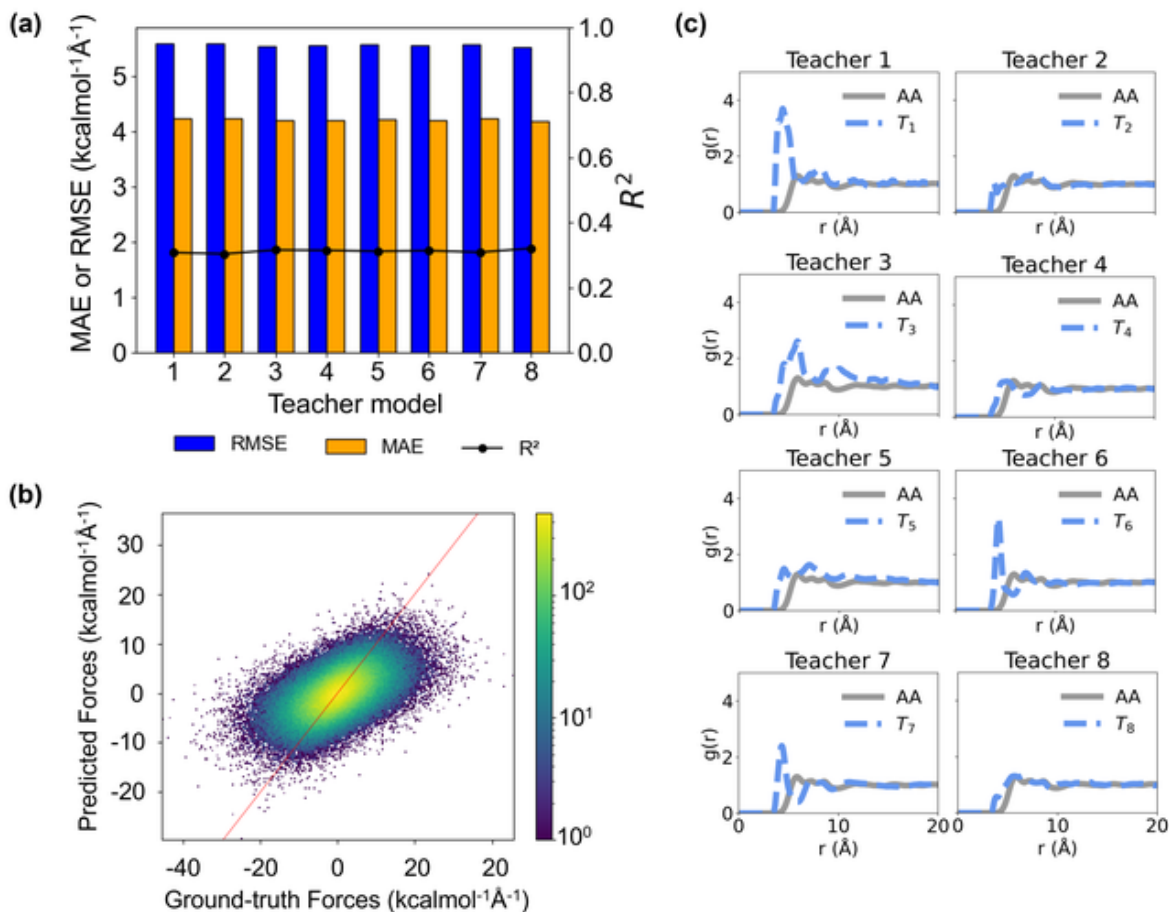


Figure 3: (a) Training metrics MAE, RMSE and R^2 of individual teacher models on ground-truth AA force targets, (b) Parity plot of the predicted versus ground-truth AA forces, (c) Urea-Urea RDF of individual teacher models in comparison to AA reference.

3.2 Effects of teacher source in student training

We trained eight teacher models (denoted as T), all using the same architecture and dataset but initialized with different random seeds. RDFs and RDF TAEs of the T models for each pair are shown in Section 3.1 of the Supporting Information (SI). CDFs and CDF TAEs for individual teachers are presented in Section 5.1 of the SI. Different seeds produce noticeably different outcomes: for example, in Figure 3, Teacher 1 exhibits a shorter-range urea-urea peak near 5 $\text{\AA}$ with markedly greater intensity ($4\times$ higher), whereas Teacher 8 more closely matches

the AA distribution.

Figure 4 reports the RDF TAE for the urea–urea pair; the RDF TAEs for the other pairs are provided in Section S3.3 of the SI. Analogously, example ADF and CDF summary TAEs appear in Figures 5 and 6, with the full ADF and CDF results given in Sections S4.1 and S5.3 of the SI. The RDF TAEs of the $\mathbf{T}$ models are large across all pairs and feature wide error bars, indicating high variability among teachers; as a rule of thumb, **RDF TAE** > 1 signifies substantial distortion of pair structure. An ADF TAE greater than 0.2° indicates a significant deviation from the reference; by this measure, the $\mathbf{T}$ models exhibit high ADF TAEs for most triplets except Cl–Cl–Cl. ADF TAEs decrease with increasing angular cutoff $r_{\max}$, because larger cutoffs sample more triplets and average out random fluctuations. CDF TAEs are similarly high, particularly for urea, with **CDF TAE** > 0.5 indicating pronounced CDF deviations.

To identify the origin of these errors, we calculated time-resolved CDFs for each species: choline CDFs are shown in Figure 7, while those of chloride and urea in Figure S2. The chloride CDF remains essentially stable over 0–300 ps. In contrast, the choline and urea CDFs change significantly, with their dominant peaks shifting to smaller cutoff distances, which reflects progressive clustering during the simulation. These results indicate that teachers trained only on noisy forces lack sufficient accuracy and stability to represent the CG system. This likely explains the prolonged potential-energy equilibration time (~ 500 ps) even when starting from an equilibrated AA-to-CG mapped configuration.

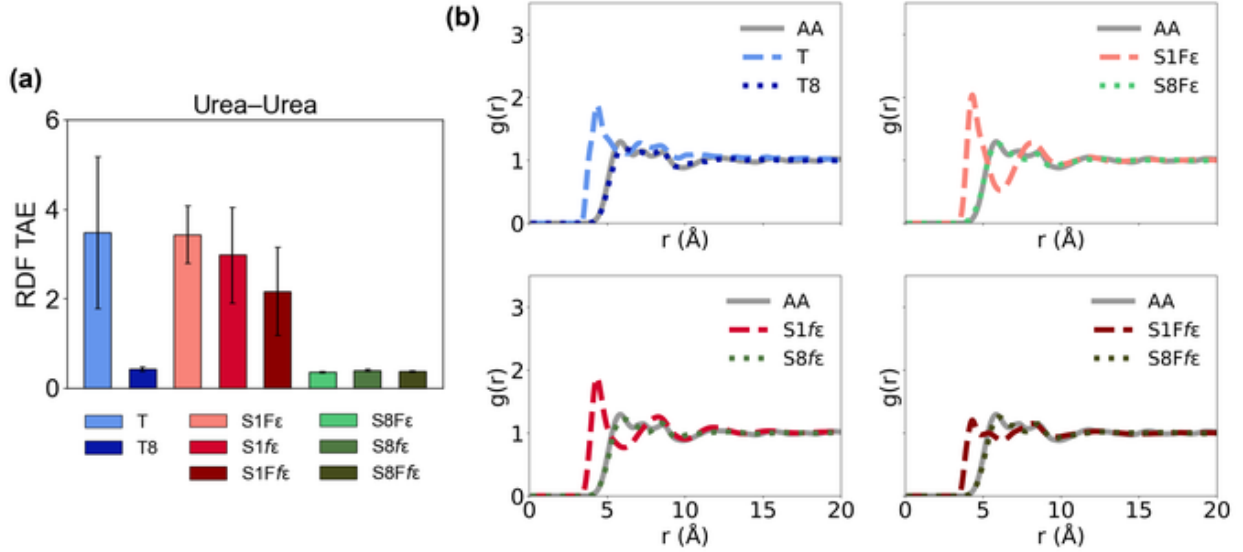


Figure 4: Comparison of urea-urea (a) RDF TAE and (b) RDF for teacher and student models (using different force targets) relative to the AA reference. Error bars denote one standard deviation over 8 replicas. Teacher (T) results were calculated as the mean of eight independent MD simulations, each performed with a teacher model trained from a unique random seed. $S1$ student variants were trained from a single teacher and then run in eight independent replicas; their reported result is the average across all eight student models, with each model trained from one of the individual teachers. $S8$ models were trained on averaged data from all eight teachers. Regarding energy data inclusion, all student models here were trained to only per-bead energies. F : ground-truth forces, f : teacher forces, E : system energy, ϵ : per-bead energies.

When we instead directly deploy the eight-teacher ensemble (denoted as $T8$) in the inference stage, the RDF, ADF, and CDF TAEs all drop substantially compared to single-teacher runs, and the simulations remain stable (no spurious clustering or time-dependent drift in the CDFs). Thus, ensembling produces more accurate and stable CG dynamics, but at the cost of roughly fivefold slower inference (Table 4).

We next trained two sets of student models: $S1$, supervised by the T model, and $S8$, supervised by the $T8$ model. We computed the same structural metrics and compared them to the AA reference. In Figures 4–6, $S1$ results are plotted in red shades and $S8$ in green shades.

The $S1$ models closely mirror the single-teacher behavior—their RDF, ADF, and CDF TAEs match or exceed those of T (for example, the urea CDF TAE in Figure 6 is higher than the

Table 4: Performance of various models on a 1000-bead DES system at 300 K and 1 bar. All benchmarks used a 2 fs time step on a 48-thread CPU. **T**: single-teacher; **T8**: eight-teacher ensemble; **S1**: single-distilled student; **S8**: ensemble-distilled student.

Model	Performance (ns/day)
T	2.66
T8	0.54
S1	2.66
S8	2.66

corresponding **T** value)—and visual inspection confirms similar clustering and aggregation. By contrast, **S8** achieves TAEs comparable to **T8** and yields stable dynamics, with time-invariant CDFs for all species (Figure 7).

In inference (Table 4), **S8** provides a clear speed advantage when compared with **T8**. Although both teacher and student share the same architecture and size, **S8** requires only one model evaluation per MD step, whereas **T8** must evaluate eight teachers and average their outputs before advancing the dynamics. Consequently, **S8** runs approximately five times faster than **T8** in our benchmarks. Overall, training the student on ensemble guidance improves both the accuracy and efficiency of ML CG simulations.

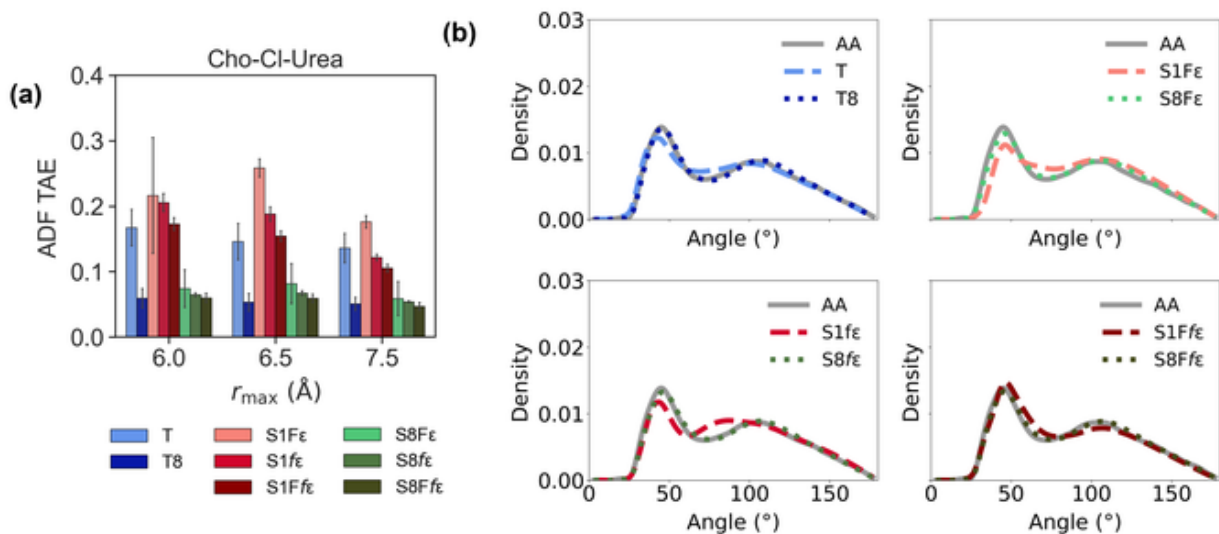


Figure 5: Comparison of Cho-Cl-Urea (a) ADF TAE at different ADF cutoff values $r_{\max}$ and (b) example ADF at $r_{\max} = 7.5$ Å at for teacher and student models relative to the AA reference (using different force targets). Error bars denote one standard deviation over 8 replicas. Teacher (**T**) results were calculated as the mean of eight independent MD simulations, each performed with a teacher model trained from a unique random seed. **S1** student variants were trained from a single teacher and then run in eight independent replicas; their reported result is the average across all eight student models, with each model trained from one of the individual teachers. **S8** models were trained on averaged data from all eight teachers. Regarding energy data inclusion, all student models here were trained to only per-atom energies. **F**: ground-truth forces, **f**: teacher forces, **E**: system energy, **ϵ** : per-bead energies.

3.3 Effects of force targets

To determine an optimal student-training protocol, we first varied the force supervision: ground-truth AA forces only (**S1F ϵ** , **S8F ϵ**), teacher-predicted forces only (**S1f ϵ** , **S8f ϵ**), and both (**S1Ff ϵ** , **S8Ff ϵ**); ϵ were included in all cases here. Performance was evaluated via RDF, ADF, and CDF TAEs in Figures 4–6 (full summary TAEs are in the SI), with individual RDFs in Section S3, ADFs at $r_{\max} = 6.0, 6.5, 7.5$ Å in Section S4, and species CDFs in Section S5 of the SI.

For the **S1** variants, using only teacher-predicted forces (**S1f ϵ**) generally reduces the RDF TAE compared to using only AA forces (**S1F ϵ**). The lowest RDF TAEs are achieved when

both force targets are combined ($\mathbf{S1Ff\epsilon}$), though the error bars indicate this improvement is modest. CDF TAEs follow the same pattern. ADF TAEs also improve with teacher forces except for Cl–Cl–Cl and Cho–Cho–Cho at $r_{\max} = 6.0$ Å, likely due to limited sampling.

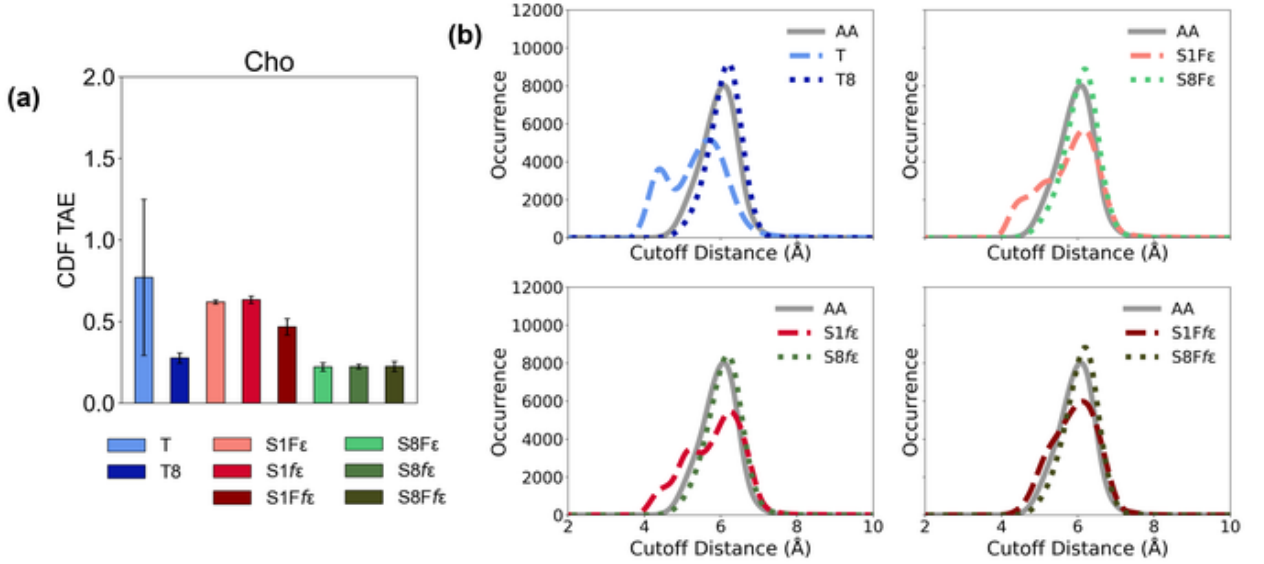


Figure 6: Comparison of Choline (Cho) (a) CDF TAE and (b) CDF for teacher and student models (using different force targets) relative to the AA reference. Error bars denote one standard deviation over 8 replicas. Teacher ($\mathbf{T}$) results were calculated as the mean of eight independent MD simulations, each performed with a teacher model trained from a unique random seed. $\mathbf{S1}$ student variants were trained from a single teacher and then run in eight independent replicas; their reported result is the average across all eight student models, with each model trained from one of the individual teachers. $\mathbf{S8}$ models were trained on averaged data from all eight teachers. Regarding energy data inclusion, all student models here were trained to only per-atom energies. $\mathbf{F}$: ground-truth forces, $\mathbf{f}$: teacher forces, $\mathbf{E}$: system energy, $\mathbf{\epsilon}$: per-bead energies.

For the $\mathbf{S8}$ variants, adding teacher forces ($\mathbf{S8Ff\epsilon}$) has minimal effect on RDF and CDF TAEs relative to $\mathbf{S8F\epsilon}$, while ADF TAEs decrease slightly across most triples, except that of Cho–Cho–Cho which remained largely unchanged.

In summary, including teacher-predicted forces benefits student accuracy for both single- and ensemble-teacher training.

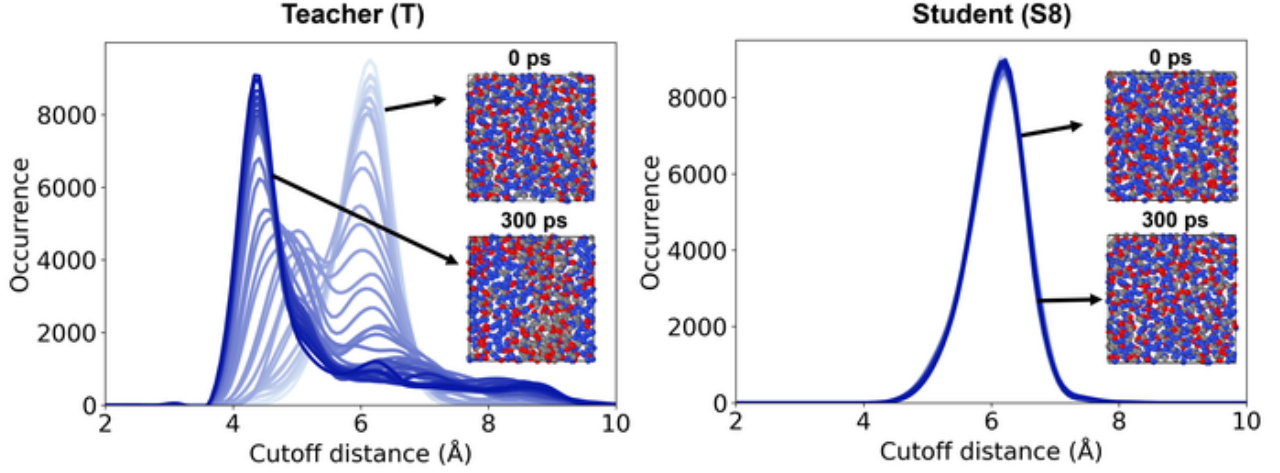


Figure 7: Temporal evolution of Choline CDFs for the teacher and ensemble-trained student models, shown at 10 ps intervals from 0 ps (lightest blue) to 300 ps (darkest blue). The teacher’s distribution gradually shifts, reflecting species aggregation, while the student models’ curves remain nearly unchanged, indicating stability over time.

3.4 Effects of energy targets

To identify effective energy targets, we evaluated four **S8** variants: no energy targets (**S8Ff**), per-bead energies only (**S8Ff ϵ**), system energy only (**S8FfE**), and both per-bead and total energies (**S8FfE ϵ**); all models here included both **F** and **f** as training targets for consistency. We only focus on **S8** models here because they were earlier shown to outperform students trained by single teachers. Figures 8 shows the RDF TAE for the Cho–Cl pair; the full RDF, ADF, and CDF profiles and their corresponding TAEs for all energy-target models are available in Sections S6–S8 of the SI.

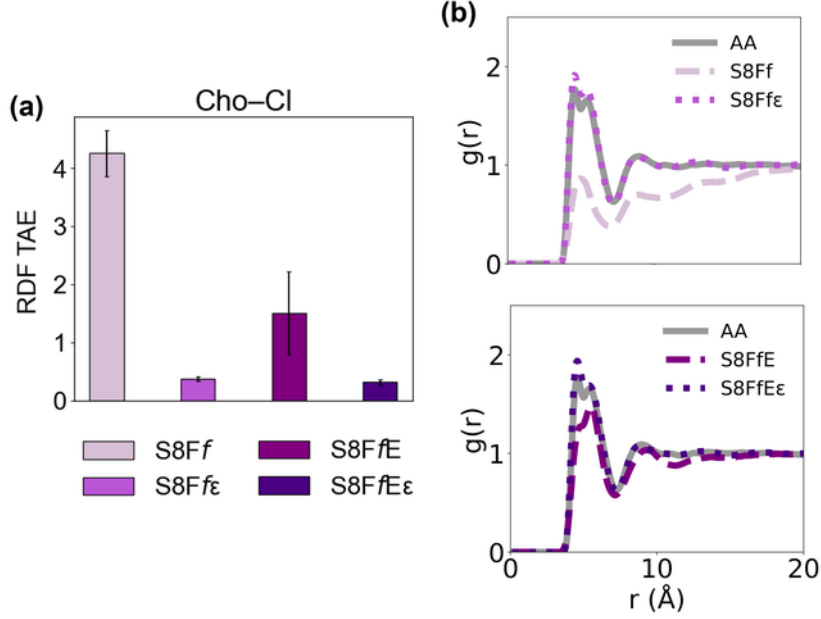


Figure 8: Comparison of Choline(Cho)-Chloride(Cl) (a) RDF TAE and (b) RDF for student models using different energy targets relative to the AA reference. Error bars denote one standard deviation over 8 replicas. **S8** models were trained on averaged data from all eight teachers. Regarding force data inclusion, all models here were trained to both ground-truth and teacher forces. **F**: ground-truth forces, **f**: teacher forces, **E**: system energy, **ε**: per-bead energies.

The RDF TAE is largest when no energy targets are used (**S8Ff**), while adding only per-bead energies (**S8Ffε**) sharply reduces the TAE and yields accurate, stable simulations across all pair distributions. Using only system energy (**S8FfE**) produces a modest increase in RDF TAE relative to **S8Ffε**, whereas combining both energies (**S8FfEε**) restores performance to the level of **S8Ffε**. ADF and CDF TAEs also follow the same qualitative trend.

Overall, per-bead energies are the critical target for stability and accuracy, with system energy alone offering little benefit, though it does no harm when used alongside per-bead energies.

4 Conclusion

Here, we present a knowledge distillation workflow for ML CG FFs to overcome noisy force labels and the difficulty of directly fitting CG energies to AA references due to its inherent inclusion of entropic contributions. Specifically, we trained initial *teacher* CG neural network potentials solely on those instantaneous, noisy forces. We then distilled both force and energy predictions from the teachers into *student* CG models and evaluated all models by comparing RDFs, ADFs, and CDFs against the AA reference using the TAE metric, as standard training metrics (MAE, RMSE, $R^2 \approx 0.35$) proved unreliable.

We trained eight teacher models with identical architecture and dataset, varied only by training seed. We observed that all teachers (in MD simulations) caused species aggregation that distorted structure, which produced large RDF, ADF, and CDF TAEs. Instead, deploying their ensemble in MD produced stable simulations with improved structural accuracy. We then trained student models on auxiliary targets from either a single teacher or an eight-teacher ensemble. Students distilled from one teacher retained its poor performance, while the ensemble-distilled student matched the ensemble’s stability and accuracy at single-model speed, roughly five times faster.

We also tested various student training protocols and found that incorporating teacher-predicted forces yields a small accuracy gain. The most impactful auxiliary target is the per-bead energy, while including system energy has little effect. Training students on ground-truth AA forces together with per-bead energies and forces predicted by the teacher ensemble therefore improves the quality of CG FFs.

Future work will apply this framework to more challenging materials such as polymers, whose high configurational variability produces much noisier forces. We will also explore successive student generations, training each new model on the auxiliary targets of its predecessor, although recent studies⁴³ suggest little gain beyond the first generation. We also plan to evalu-

ate how network size and architecture affect the trade-off between accuracy and speed. While our main aim here is to improve CG model stability and accuracy under noisy forces, future work may also address network compression. We anticipate that this approach can deliver accurate, efficient CG FFs to study phenomena at higher length and time scales.

Data and Software Availability

We provide all simulation files used in this study: GROMACS inputs for training-data generation and LAMMPS scripts for CG simulations are listed in Section S9 of the SI. The FF and structure files are available in the OPLS-DES repository,⁵¹ and the model-training scripts can be found in the open-source `hippynn` repository.⁵⁹

Supporting Information

Temporal evolution of model observables, training metrics, two- (RDF), three-(ADF) and many-(CDF) body evaluations of teacher and student models with varying force and energy targets, input scripts for reproducing workflow.

Author Information

Corresponding Authors

Emily Shinkle - *Computing and Artificial Intelligence Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, United States*; Email: eshinkle@lanl.gov

Nicholas Lubbers - *Computing and Artificial Intelligence Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, United States*; Email: nlubbers@lanl.gov

Authors

Feranmi V. Olowookere – *Department of Chemical and Biological Engineering, The University of Alabama, Tuscaloosa, AL 35487-0203, United States.*

Sakib Matin - *Theoretical Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, United States.*

Aleksandra Pachalieva - *Earth and Environmental Sciences Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, United States; Center for Nonlinear Studies, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, United States.*

Author Contributions

F.V.O. Software, validation, investigation, visualization, data curation, writing—original draft.

S.M. Conceptualization, methodology, validation, writing—review and editing.

A.P. Conceptualization, validation, writing—review and editing.

N.L. Conceptualization, validation, methodology, software, supervision, project administration, resources, Funding acquisition, writing—review and editing.

E.S. Conceptualization, validation, methodology, software, supervision, project administration, resources, Funding acquisition, writing—review and editing.

Notes

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Research presented in this article was supported by the National Security Education Center (NSEC) Informational Science and Technology Institute (ISTI) using the Laboratory Directed Research and Development program of Los Alamos National Laboratory project number 20240479CR-IST as part of the Applied Machine Learning Summer Research Fellowship program. This research used resources provided by the Darwin testbed at LANL which is funded by the Computational Systems and Software Environments subprogram of LANL's Advanced Simulation and Computing program. LANL is operated by the Triad National Security, LLC, for the National Nuclear Security Administration of the U.S. Department of Energy (contract no. 89233218NCA000001). We are grateful to Heath Turner and Galen Craven for insightful discussions and suggestions.

References

- (1) Allen, M. P.; Frenkel, D.; Talbot, J. Molecular dynamics simulation using hard particles. *Comput. Phys. Rep.* **1989**, *9*, 301–353.
- (2) Hansen, J.-P.; McDonald, I. R., *Theory of simple liquids: with applications to soft matter*; Academic press: 2013.
- (3) Tuckerman, M. E.; Martyna, G. J. Understanding modern molecular dynamics: Techniques and applications. *J. Phys. Chem. B* **2000**, *104*, 159–178.
- (4) Hollingsworth, S. A.; Dror, R. O. Molecular dynamics simulation for all. *Neuron* **2018**, *99*, 1129–1143.
- (5) Sanbonmatsu, K.; Tung, C.-S. High performance computing in biology: multimillion atom simulations of nanoscale systems. *J. Struct. Biol.* **2007**, *157*, 470–480.
- (6) Joshi, S. Y.; Deshmukh, S. A. A review of advancements in coarse-grained molecular dynamics simulations. *Mol. Simul.* **2021**, *47*, 786–803.
- (7) Husic, B. E.; Charron, N. E.; Lemm, D.; Wang, J.; Pérez, A.; Majewski, M.; Krämer, A.; Chen, Y.; Olsson, S.; De Fabritiis, G., et al. Coarse graining molecular dynamics with graph neural networks. *J. Chem. Phys.* **2020**, *153*, 194101.
- (8) Voth, G. A., *Coarse-Graining of Condensed Phase and Biomolecular Systems*; Voth, G. A., Ed.; CRC Press: 2008.
- (9) Noid, W. G. Perspective: Coarse-grained models for biomolecular systems. *J. Chem. Phys.* **2013**, *139*, 090901.
- (10) Marrink, S. J.; Risselada, J.; Yefimov, S.; Tieleman, D. P.; de Vries, A. H. The MARTINI force field: coarse grained model for biomolecular simulations. *J. Phys. Chem. B* **2007**, *111*, 7812–7824.

- (11) Marrink, S. J.; Tieleman, D. P. Perspective on the Martini model. *Chem. Soc. Rev.* **2013**, *42*, 6801–6822.
- (12) Alessandri, R.; Souza, P. C. T.; Thallmair, S.; Melo, M. N.; de Vries, A. H.; Marrink, S. J. Pitfalls of the Martini model. *J. Chem. Theory Comput.* **2019**, *15*, 5448–5460.
- (13) Alessandri, R.; Souza, P. C.; Thallmair, S.; Melo, M. N.; De Vries, A. H.; Marrink, S. J. Pitfalls of the Martini model. *J. Chem. Theory Comput.* **2019**, *15*, 5448–5460.
- (14) Marrink, S. J.; Monticelli, L.; Melo, M. N.; Alessandri, R.; Tieleman, D. P.; Souza, P. C. Two decades of Martini: Better beads, broader scope. *Wiley Interdiscip. Rev.:Comput. Mol. Sci.* **2023**, *13*, e1620.
- (15) Ercolessi, F.; Adams, J. B. Interatomic potentials from first-principles calculations: the force-matching method. *Europhys. Lett.* **1994**, *26*, 583–588.
- (16) Izvekov, S.; Voth, G. A. A multiscale coarse-graining method for biomolecular systems. *J. Phys. Chem. B* **2005**, *109*, 2469–2473.
- (17) Reith, D.; Pütz, M.; Müller-Plathe, F. Deriving effective mesoscale potentials from atomistic simulations. *J. Comput. Chem.* **2003**, *24*, 1624–1636.
- (18) Shell, M. S. The relative entropy is fundamental to multiscale and inverse thermodynamic problems. *J. Chem. Phys.* **2008**, *129*, 144108.
- (19) Henderson, R. L. A uniqueness theorem for fluid pair correlation functions. *Phys. Lett. A* **1974**, *49*, 197–198.
- (20) Shinkle, E.; Pachaliev, A.; Bahl, R.; Matin, S.; Gifford, B.; Craven, G. T.; Lubbers, N. Thermodynamic transferability in coarse-grained force fields using graph neural networks. *J. Chem. Theory Comput.* **2024**, *20*, 10524–10539.
- (21) Wang, W.; Ribeiro, J. M. L.; Tiwary, P. CGnet: deep learning for coarse-grained molecular simulations. *J. Chem. Phys.* **2019**, *151*, 124113.

- (22) Tang, Z.; Zhang, Z.; Liang, H. Deep learning molecular coarse-graining using virtual-site based graph convolutional networks. *J. Chem. Phys.* **2020**, *152*, 124114.
- (23) Lemke, T.; Peter, C. Neural network based prediction of coarse-grained molecular dynamics potentials. *J. Chem. Theory Comput.* **2017**, *13*, 6213–6225.
- (24) Jin, J.; Schweizer, K. S.; Voth, G. A. Understanding dynamics in coarse-grained models. I. Universal excess entropy scaling relationship. *J. Chem. Phys.* **2023**, *158*, 034103, <https://doi.org/10.1063/5.0116299>.
- (25) Palma Banos, M.; Popov, A. V.; Hernandez, R. Representability and Dynamical Consistency in Coarse-Grained Models. *J. Phys. Chem. B* **2024**, *128*, 1506–1514.
- (26) Han, Y.; Jin, J.; Voth, G. A. Constructing many-body dissipative particle dynamics models of fluids from bottom-up coarse-graining. *J. Chem. Phys.* **2021**, *154*.
- (27) Meinel, M. K.; Müller-Plathe, F. Predicting the artificial dynamical acceleration of binary hydrocarbon mixtures upon coarse-graining with roughness volumes and simple averaging rules. *J. Chem. Phys.* **2024**, *160*.
- (28) Bag, S.; Meinel, M. K.; Müller-Plathe, F. Toward a mobility-preserving coarse-grained model: A data-driven approach. *J. Chem. Theory Comput.* **2022**, *18*, 7108–7120.
- (29) Unke, O. T.; Chmiela, S.; Sauceda, H. E.; Gastegger, M.; Poltavsky, I.; Schutt, K. T.; Tkatchenko, A.; Müller, K.-R. Machine learning force fields. *Chem. Rev.* **2021**, *121*, 10142–10186.
- (30) Wu, S.; Yang, X.; Zhao, X.; Li, Z.; Lu, M.; Xie, X.; Yan, J. Applications and advances in machine learning force fields. *Journal of Chemical Information and Modeling* **2023**, *63*, 6972–6985.
- (31) Behler, J. Perspective: Machine learning potentials for atomistic simulations. *J. Chem. Phys.* **2016**, *145*, 170901.

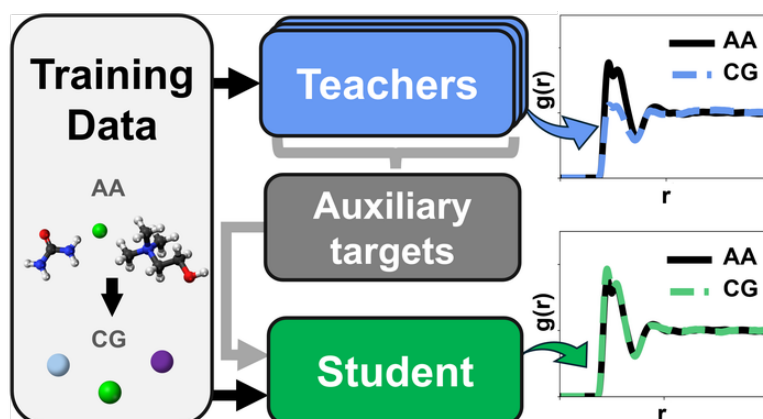
- (32) Kulichenko, M.; Smith, J. S.; Nebgen, B.; Li, Y. W.; Fedik, N.; Boldyrev, A. I.; Lubbers, N.; Barros, K.; Tretiak, S. The rise of neural networks for materials and chemical dynamics. *J. Phys. Chem. Lett.* **2021**, *12*, 6227–6243.
- (33) Vaikuntanathan, S.; Voth, G. A. Numerical methods to obtain potentials of mean force via coarse-graining. *J. Chem. Phys.* **2009**, *130*, 094102.
- (34) Wang, W.; Ribeiro, J. M. L.; Tiwary, P. Machine learning of coarse-grained molecular dynamics force fields. *Proc. Natl. Acad. Sci. U. S. A.* **2019**, *116*, 20750–20759.
- (35) Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* **2015**.
- (36) Buciluă, C.; Caruana, R.; Niculescu-Mizil, A. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006, pp 535–541.
- (37) Ba, J.; Caruana, R. In *Advances in Neural Information Processing Systems*, 2014; Vol. 27.
- (38) Martin, S.; Allen, A. E. A.; Shinkle, E.; Pachalieva, A.; Craven, G. T.; Nebgen, B.; Smith, J. S.; Messerly, R.; Li, Y. W.; Tretiak, S.; Barros, K.; Lubbers, N. Teacher-student training improves the accuracy and efficiency of machine learning interatomic potentials. *Digital Discovery* **2025**, *4*, 2502–2511.
- (39) Ekström Kelvinius, F.; Georgiev, D.; Toshev, A.; Gasteiger, J. Accelerating molecular graph neural networks via knowledge distillation. *Advances in Neural Information Processing Systems* **2023**, *36*, 25761–25792.
- (40) Amin, I.; Raja, S.; Krishnapriyan, A. Towards fast, specialized machine learning force fields: Distilling foundation models via energy Hessians. *arXiv preprint arXiv:2501.09009* **2025**.
- (41) Yang, K. et al. Towards learning generalizable interatomic potentials with knowledge distillation. *J. Chem. Theory Comput.* **2022**, *18*, 3410–3420.

- (42) Zhou, X. et al. Distilling neural potentials: compressing ensembles into single models. *J. Chem. Phys.* **2023**, *158*, 234104.
- (43) Matin, S.; Shinkle, E.; Pimonova, Y.; Craven, G. T.; Pachaliev, A.; Li, Y. W.; Barros, K.; Lubbers, N. Ensemble Knowledge Distillation for Machine Learning Interatomic Potentials. *arXiv preprint arXiv:2503.14293* **2025**.
- (44) Dietterich, T. G. In *International Workshop on Multiple Classifier Systems*, Springer: 2000, pp 1–15.
- (45) Lakshminarayanan, B.; Pritzel, A.; Blundell, C. In *Advances in Neural Information Processing Systems*, ed. by Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; Garnett, R., Curran Associates, Inc.: 2017; Vol. 30, https://proceedings.neurips.cc/paper_files/paper/2017/file/9ef2ed4b7fd2c810847ffa5fa85bce3Paper.pdf.
- (46) Durumeric, A. E.; Chen, Y.; Noé, F.; Clementi, C. Learning data efficient coarse-grained molecular dynamics from forces and noise. *arXiv preprint arXiv:2407.01286* **2024**.
- (47) Rosenberger, D.; Barros, K.; Germann, T. C.; Lubbers, N. Machine learning of consistent thermodynamic models using automatic differentiation. *Phys. Rev. E* **2022**, *105*, 045301, <https://link.aps.org/doi/10.1103/PhysRevE.105.045301>.
- (48) Lubbers, N.; Smith, J. S.; Barros, K. Hierarchical modeling of molecular energies using a deep neural network. *J. Chem. Phys.* **2018**, *148*, 241715.
- (49) Chigayev, M.; Smith, J. S.; Anaya, S.; Nebgen, B.; Bettencourt, M.; Barros, K.; Lubbers, N. Lightweight and effective tensor sensitivity for atomistic neural networks. *J. Chem. Phys.* **2023**, *158*, 184108.
- (50) Lindahl, E.; Hess, B.; Van Der Spoel, D. GROMACS 3.0: a package for molecular simulation and trajectory analysis. *Molecular modeling annual* **2001**, *7*, 306–317.

- (51) Doherty, B.; Acevedo, O. OPLS force field for choline chloride-based deep eutectic solvents. *J. Phys. Chem. B* **2018**, *122*, 9982–9993.
- (52) Zhong, X.; Velez, C.; Acevedo, O. Partial charges optimized by genetic algorithms for deep eutectic solvent simulations. *J. Chem. Theory Comput.* **2021**, *17*, 3078–3087.
- (53) Martínez, L.; Andrade, R.; Birgin, E. G.; Martínez, J. M. PACKMOL: A package for building initial configurations for molecular dynamics simulations. *J. Comput. Chem.* **2009**, *30*, 2157–2164.
- (54) Zhang, Y.; Poe, D.; Heroux, L.; Squire, H.; Doherty, B. W.; Long, Z.; Dadmun, M.; Gurkan, B.; Tuckerman, M. E.; Maginn, E. J. Liquid structure and transport properties of the deep eutectic solvent ethaline. *J. Phys. Chem. B* **2020**, *124*, 5251–5264.
- (55) Nosé, S. A unified formulation of the constant temperature molecular dynamics methods. *J. Chem. Phys.* **1984**, *81*, 511–519.
- (56) Hess, B.; Bekker, H.; Berendsen, H. J.; Fraaije, J. G. LINCS: A linear constraint solver for molecular simulations. *J. Comput. Chem.* **1997**, *18*, 1463–1472.
- (57) Darden, T.; York, D.; Pedersen, L., et al. Particle mesh Ewald: An $N \log(N)$ method for Ewald sums in large systems. *J. Chem. Phys.* **1993**, *98*, 10089–10089.
- (58) Krämer, A.; Durumeric, A. E.; Charron, N. E.; Chen, Y.; Clementi, C.; Noé, F. Statistically optimal force aggregation for coarse-graining molecular dynamics. *J. Phys. Chem. Lett.* **2023**, *14*, 3970–3979.
- (59) Lubbers, N.; Byrd, D.; Gonzales, M.; Nebgen, B.; Li, Y. W.; Li, X.; Zhou, G.; Anaya, S.; Martin, S.; Shinkle, E.; Taylor, M. G.; Janssen, J.; Kaymak, C.; Zhang, S.; Chigaev, M. hippynn: Hierarchical Interaction Potential Neural Network Library GitHub repository, <https://github.com/lanl/hippynn> (accessed 07/11/2025).
- (60) Brehm, M.; Thomas, M.; Gehrke, S.; Kirchner, B. TRAVIS—A free analyzer for trajectories from molecular simulation. *J. Chem. Phys.* **2020**, *152*, 164105.

- (61) Chodera, J. D. A simple method for automated equilibration detection in molecular simulations. *J. Chem. Theory Comput.* **2016**, *12*, 1799–1805.
- (62) Fritz, D.; Koschke, K.; Harmandaris, V. A.; van der Vegt, N. F.; Kremer, K. Multiscale modeling of soft matter: scaling of dynamics. *Phys. Chem. Chem. Phys.* **2011**, *13*, 10412–10420.
- (63) Jin, J.; Schweizer, K. S.; Voth, G. A. Understanding dynamics in coarse-grained models. I. Universal excess entropy scaling relationship. *J. Chem. Phys.* **2023**, *158*, 034103.
- (64) Frömbgen, T.; Blasius, J.; Alizadeh, V.; Chaumont, A.; Brehm, M.; Kirchner, B. Cluster analysis in liquids: A novel tool in TRAVIS. *J. Chem. Inf. Model.* **2022**, *62*, 5634–5644.
- (65) Krämer, A.; Durumeric, A. E. P.; Charron, N. E.; Chen, Y.; Clementi, C.; Noé, F. Statistically Optimal Force Aggregation for Coarse-Graining Molecular Dynamics. *J. Phys. Chem. Lett.* **2023**, *14*, 3970–3979.

TOC Graphic



Supporting Information

S1 Temporal evolution of model observables

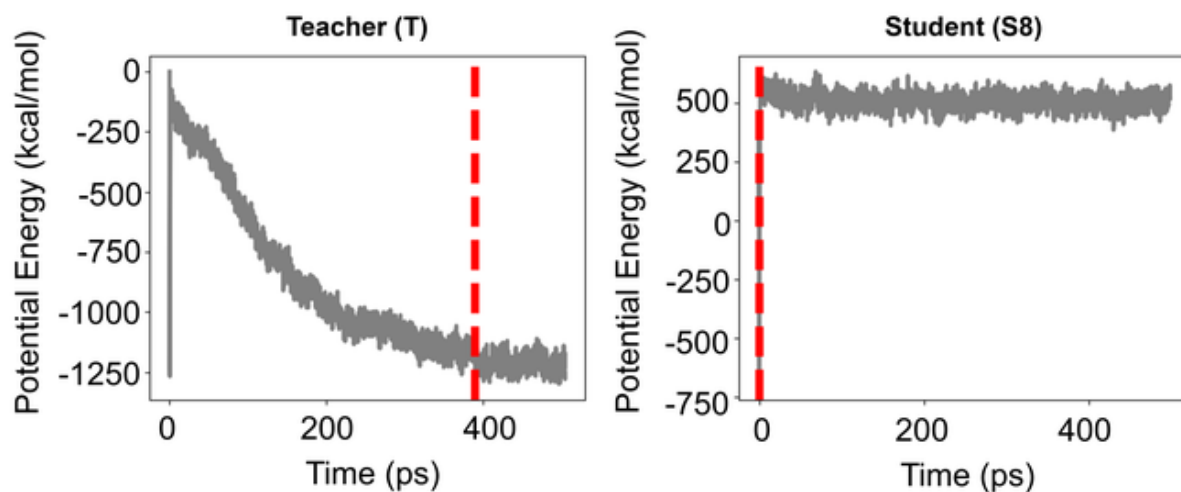


Figure S1: Time evolution of potential energy for a single selected replica, from the teacher or ensemble-trained student models. Solid red lines indicate the detection of steady-state according to the pymbar heuristic.

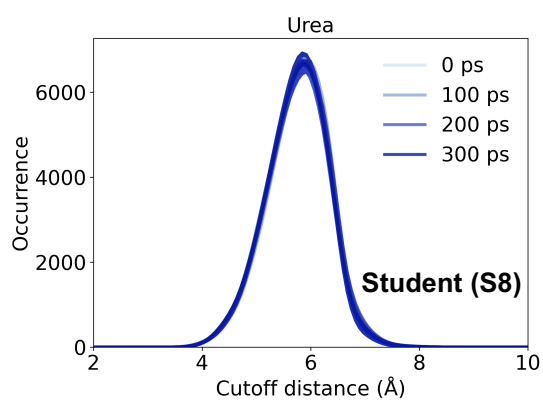
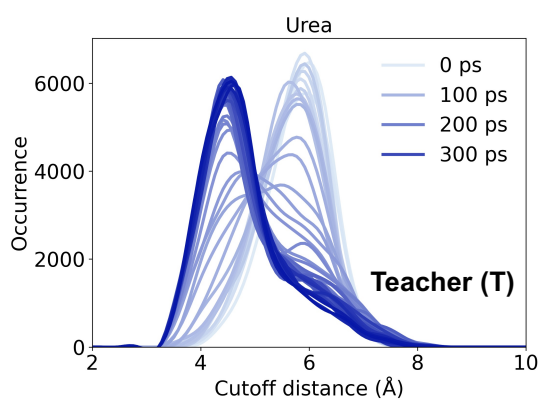
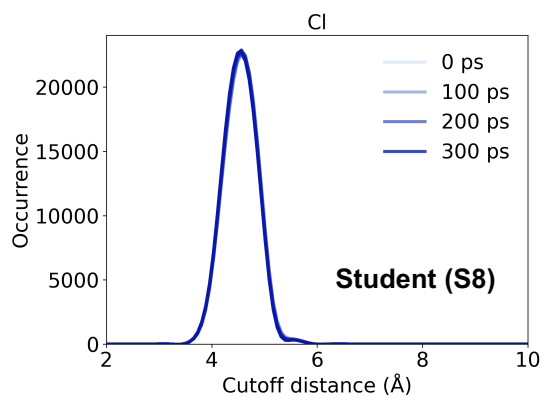
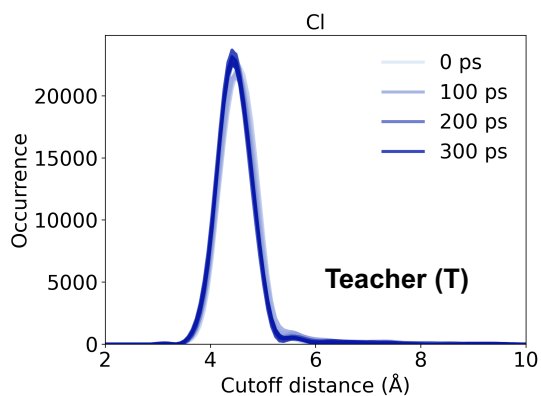


Figure S2: Temporal evolution of Chloride and Urea CDFs for the teacher and ensemble-trained student models, shown at 10 ps intervals from 0 ps (lightest blue) to 300 ps (darkest blue).

S2 Training Metrics

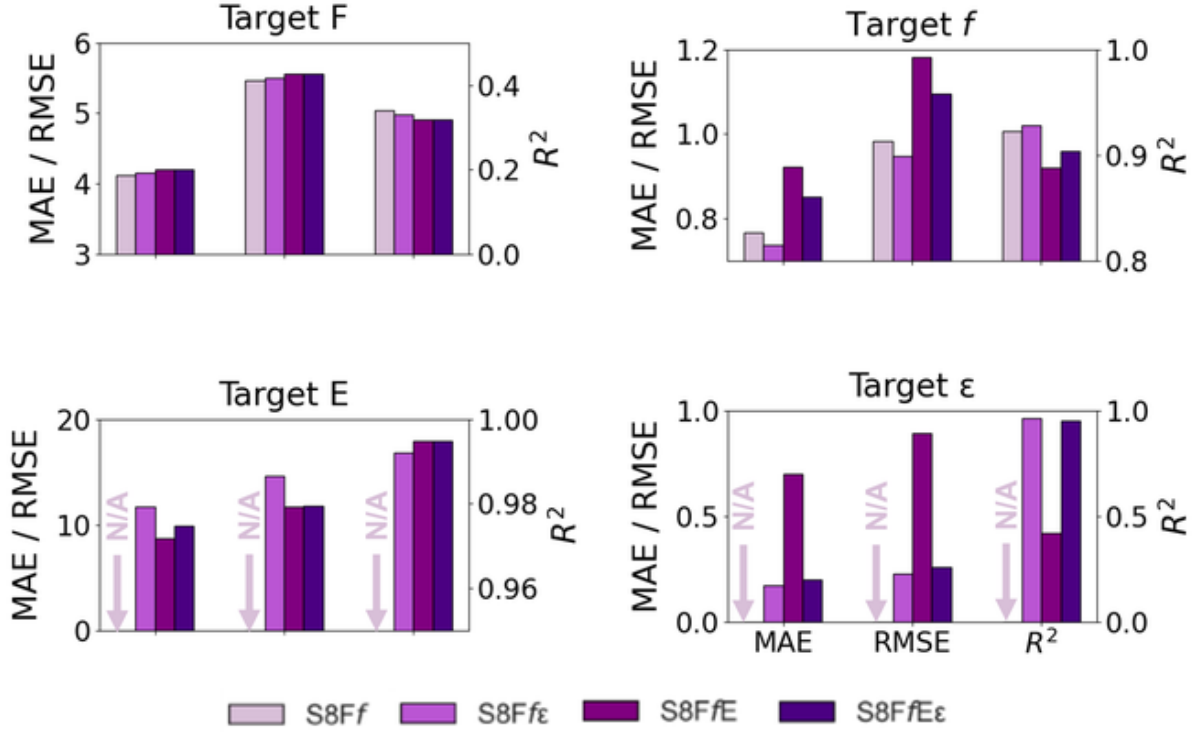


Figure S3: Training metrics MAE, RMSE and R^2 of student models (trained from an ensemble of 8 teachers). **F**: ground-truth forces, **f**: teacher forces, **E**: total energy, **ε**: per-bead energies.

S3 Two-body evaluation with varying force targets

S3.1 RDF TAE and RDFs of teacher models

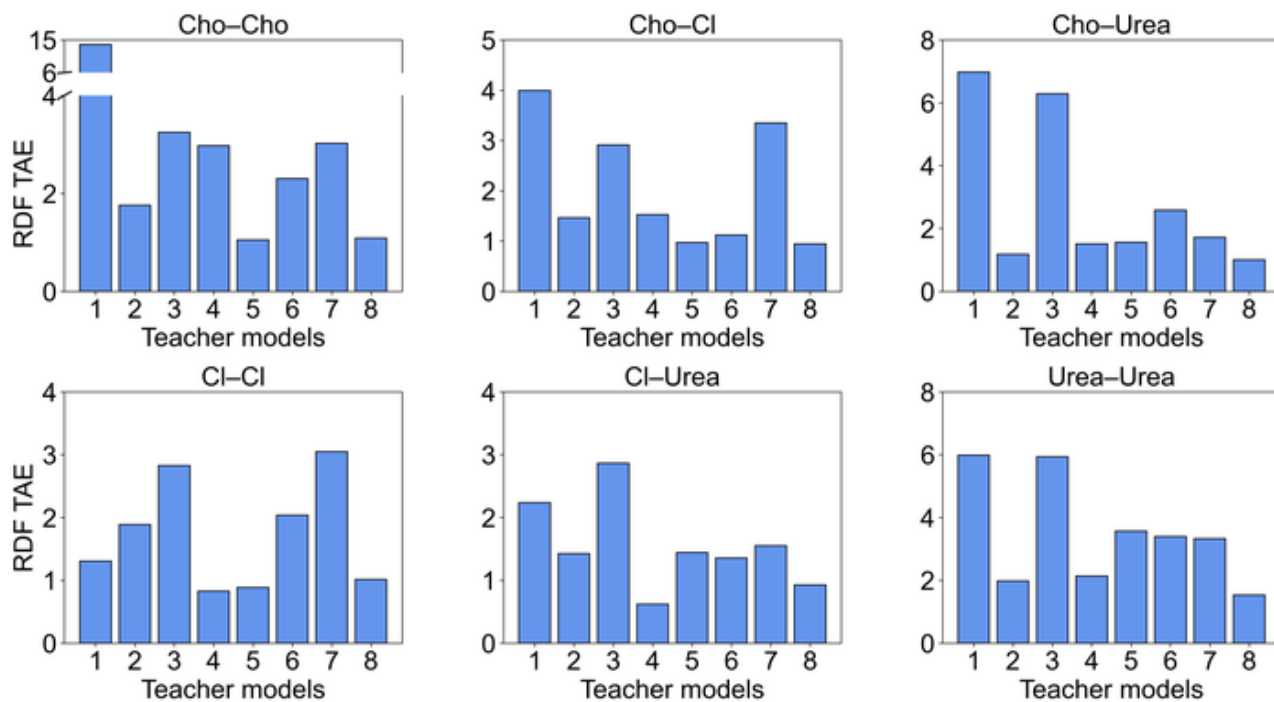


Figure S4: Comparison of RDF TAE of each pair, shown for each teacher model trained on the same data and architecture but with different random seeds.

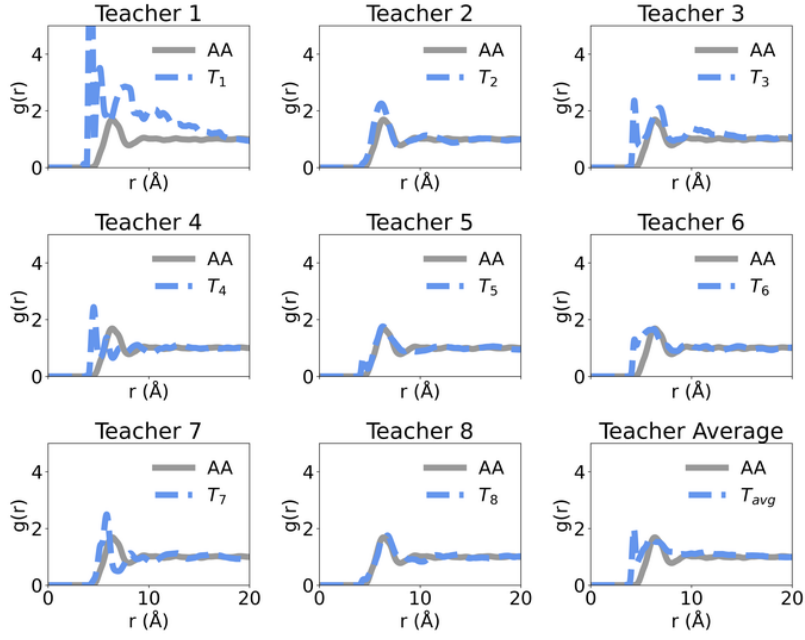


Figure S5: Cho-Cho RDFs, shown for each teacher model trained on the same data and architecture but with different random seeds.

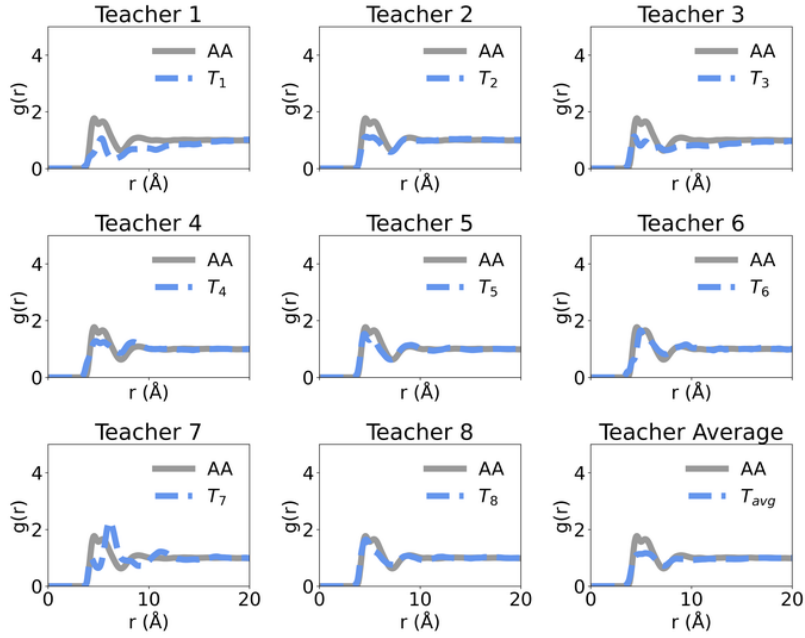


Figure S6: Cho-Cl RDFs, shown for each teacher model trained on the same data and architecture but with different random seeds.

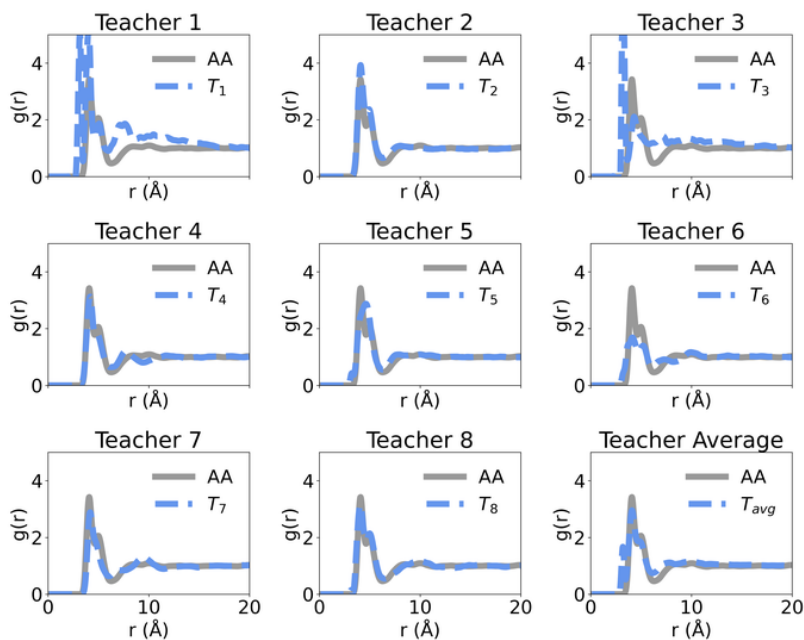


Figure S7: Cho-Urea RDFs, shown for each teacher model trained on the same data and architecture but with different random seeds.

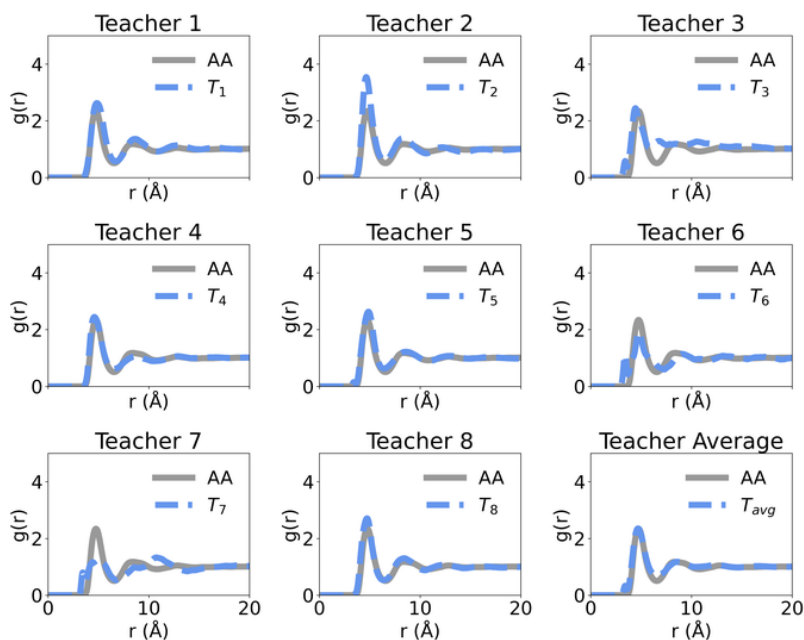


Figure S8: Cl-Cl RDFs, shown for each teacher model trained on the same data and architecture but with different random seeds.

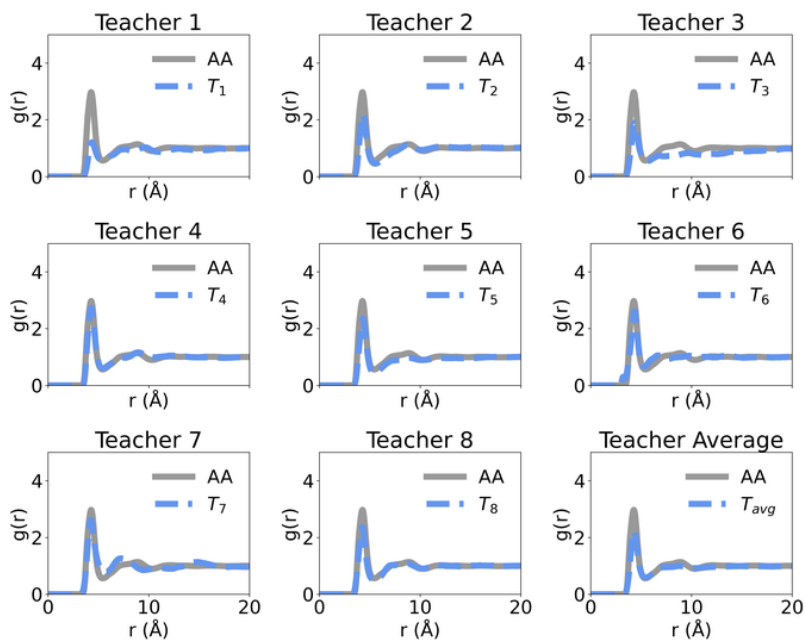


Figure S9: Cl-Urea RDFs, shown for each teacher model trained on the same data and architecture but with different random seeds.

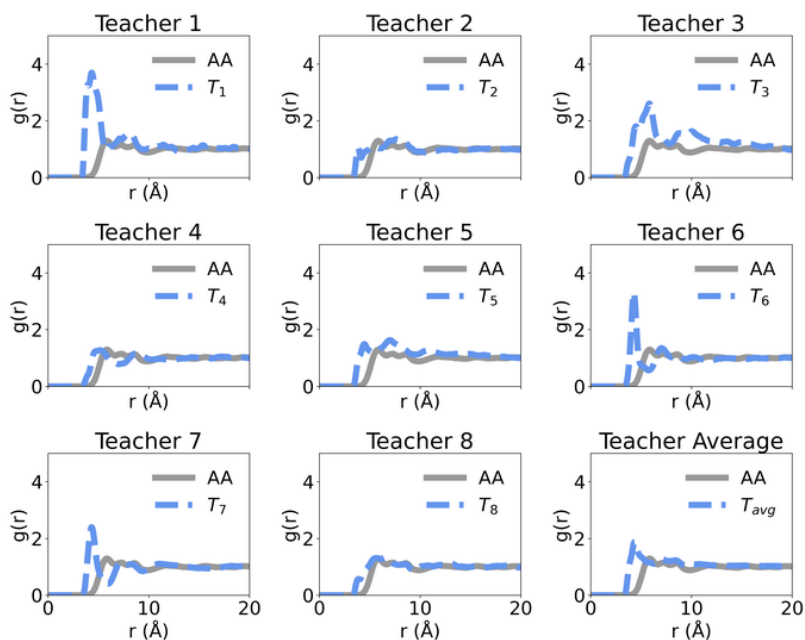


Figure S10: Urea-Urea RDFs, shown for each teacher model trained on the same data and architecture but with different random seeds.

S3.2 RDF TAE of student models with varying force targets

Figure S11 shows the RDF TAEs for each of the eight **S1** models, each trained from its corresponding teacher. Figure S12 presents the overall RDF TAE aggregated across all pairs.

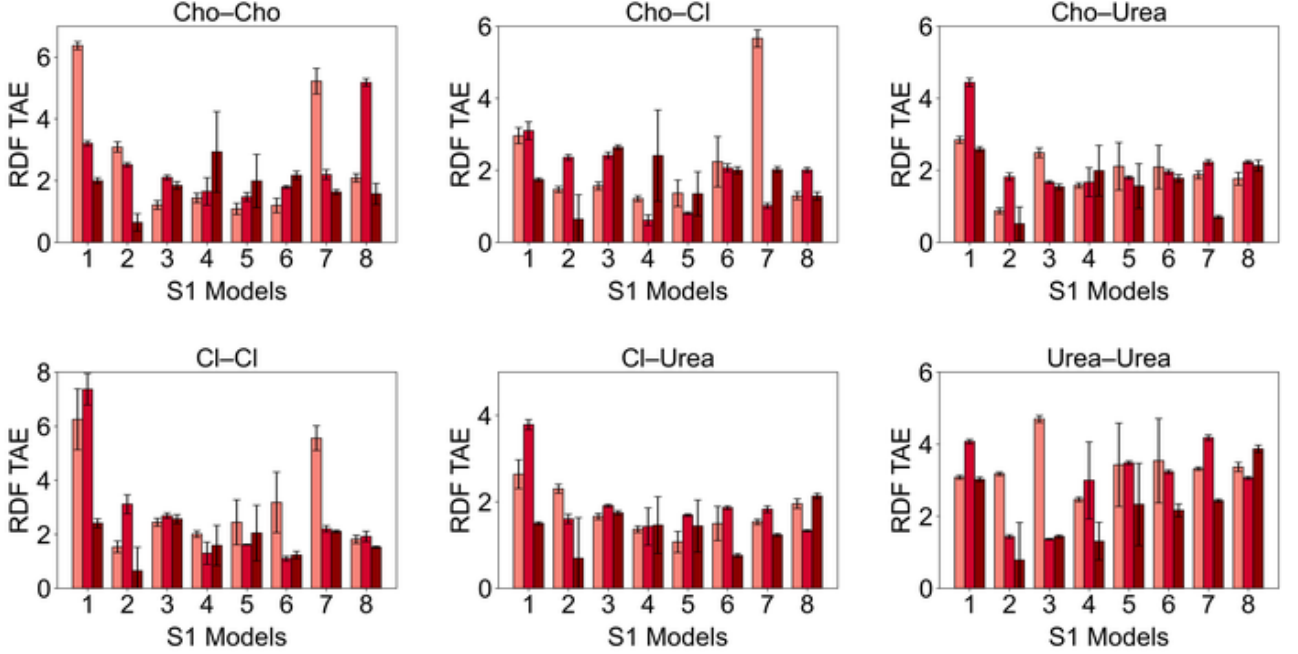


Figure S11: Comparison of RDF TAE for student models (using different force targets) relative to the AA reference. Each model was trained from respective individual teachers 1-8. Error bars denote one standard deviation over 8 replicas. Regarding energy data inclusion, the models were trained to only per-atom energies. F : ground-truth forces, f : teacher forces, E : total energy, ϵ : per-bead energies.

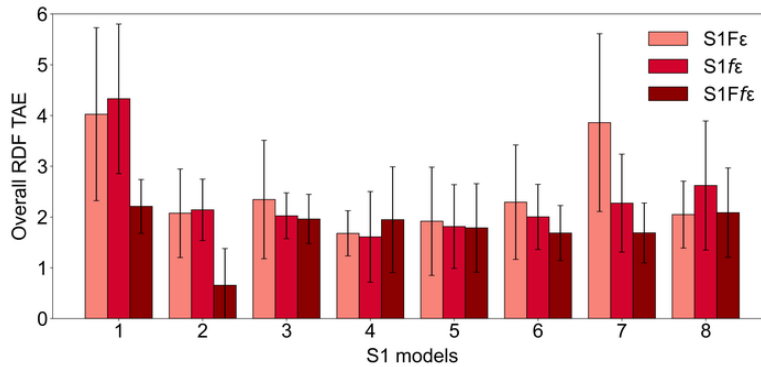


Figure S12: Comparison of overall RDF TAE for student models (using different force targets) relative to the AA reference. Each model was trained from respective individual teachers 1-8. For each model, the overall RDF TAE is calculated as the sum of the RDF TAEs of all (6) pairs in the system. Error bars denote one standard deviation over 8 replicas. Regarding energy data inclusion, the models were trained to only per-atom energies. F : ground-truth forces, f : teacher forces, E : total energy, ϵ : per-bead energies.

S3.3 Summary RDF TAE and RDFs of teacher and student models with varying force targets

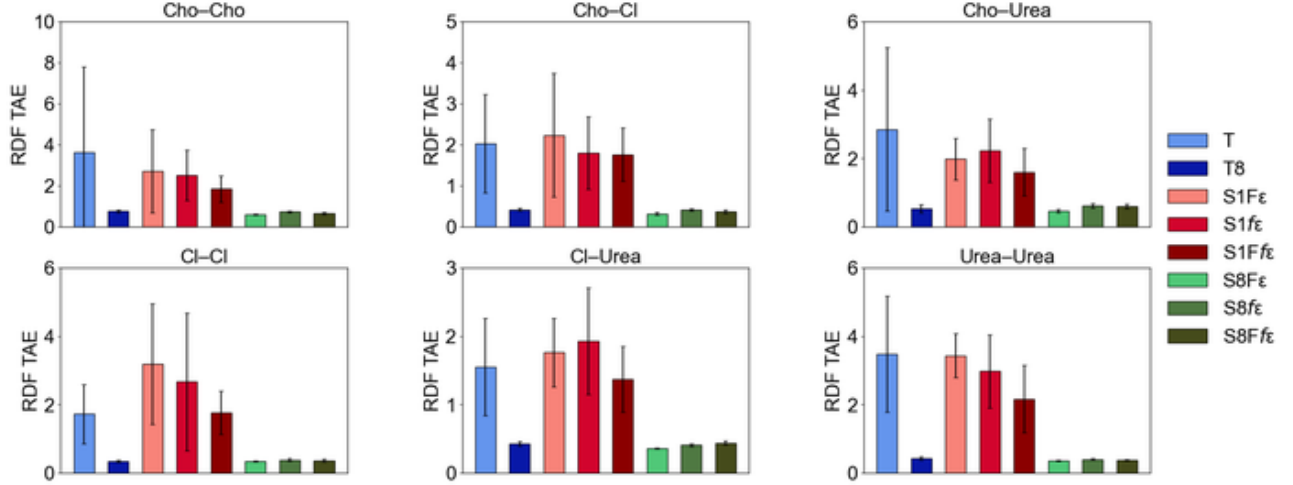


Figure S13: Comparison of RDF TAE for teacher and student models (using different force targets) relative to the AA reference. Error bars denote one standard deviation over 8 replicas. Teacher (T) results were calculated as the mean of eight independent MD simulations, each performed with a teacher model trained from a unique random seed. $S1$ student variants are trained from a single teacher and then run in eight independent replicas; their reported result is the average across all eight student models, with each model trained from one of the individual teachers. $S8$ models are trained on averaged data from all eight teachers. Regarding energy data inclusion, student models here were trained to only per-bead energies. F : ground-truth forces, f : teacher forces, E : total energy, ϵ : per-bead energies.

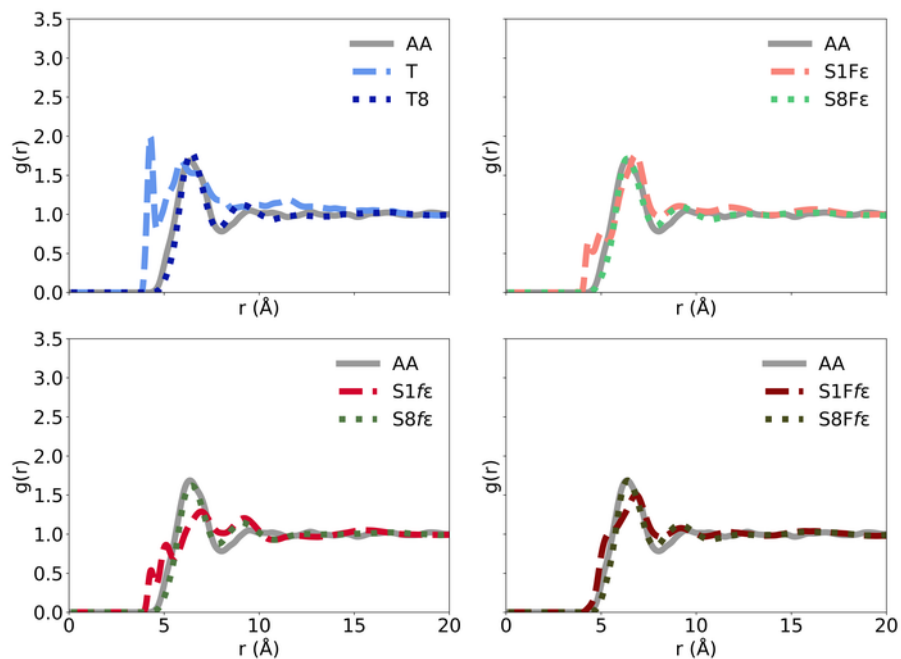


Figure S14: Cho-Cho RDFs, shown for all teacher and student models using different force targets.

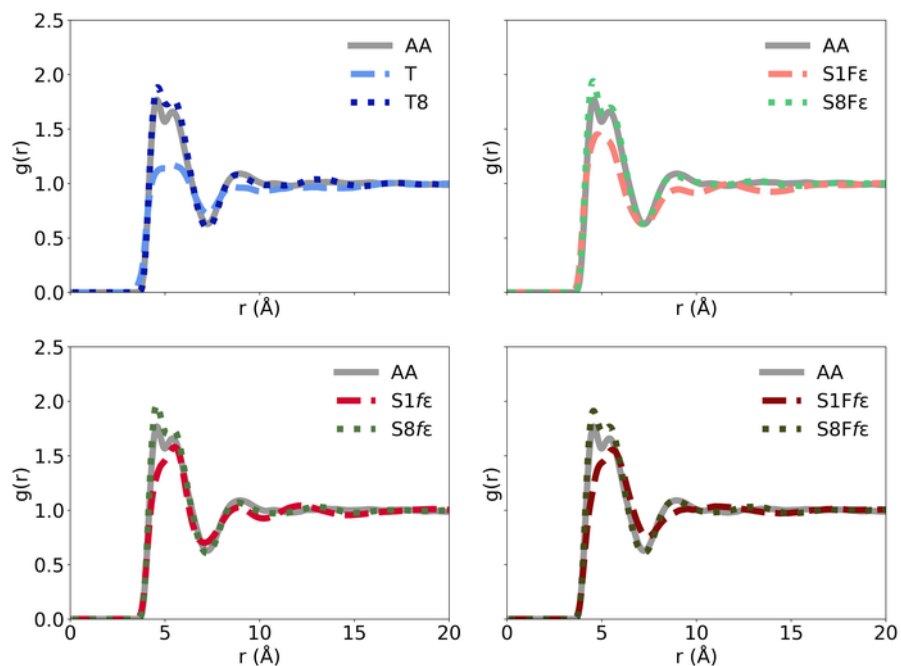


Figure S15: Cho-Cl RDFs, shown for all teacher and student models using different force targets.

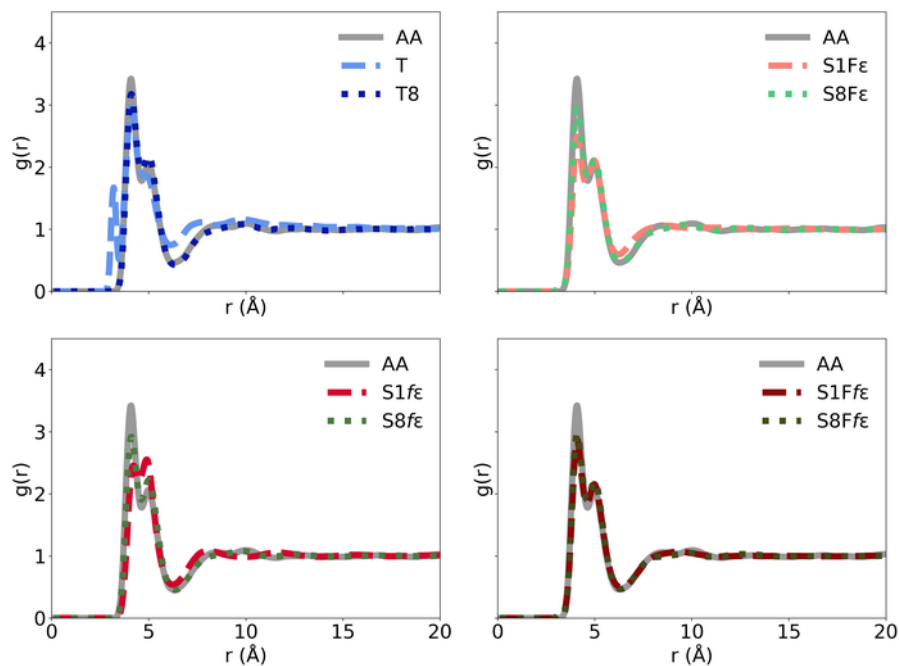


Figure S16: Cho-Urea RDFs, shown for all teacher and student models using different force targets.

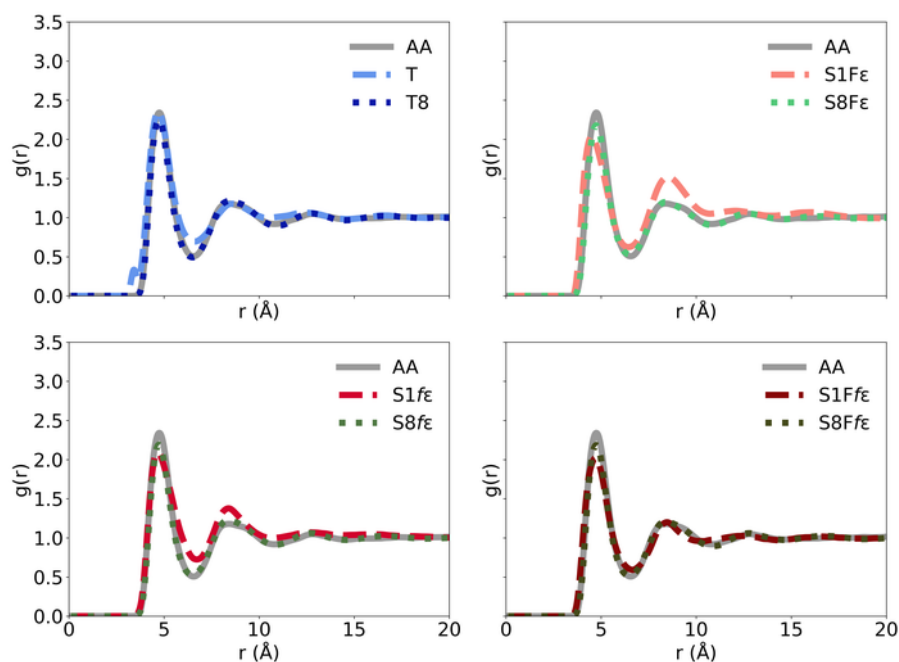


Figure S17: Cl-Cl RDFs, shown for all teacher and student models using different force targets.

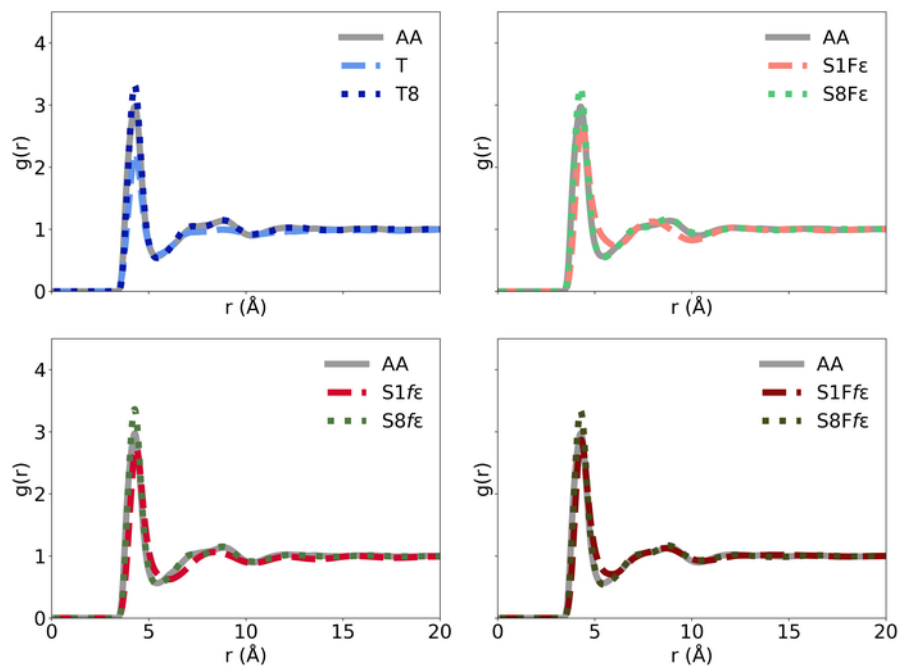


Figure S18: Cl-Urea RDFs, shown for all teacher and student models using different force targets.

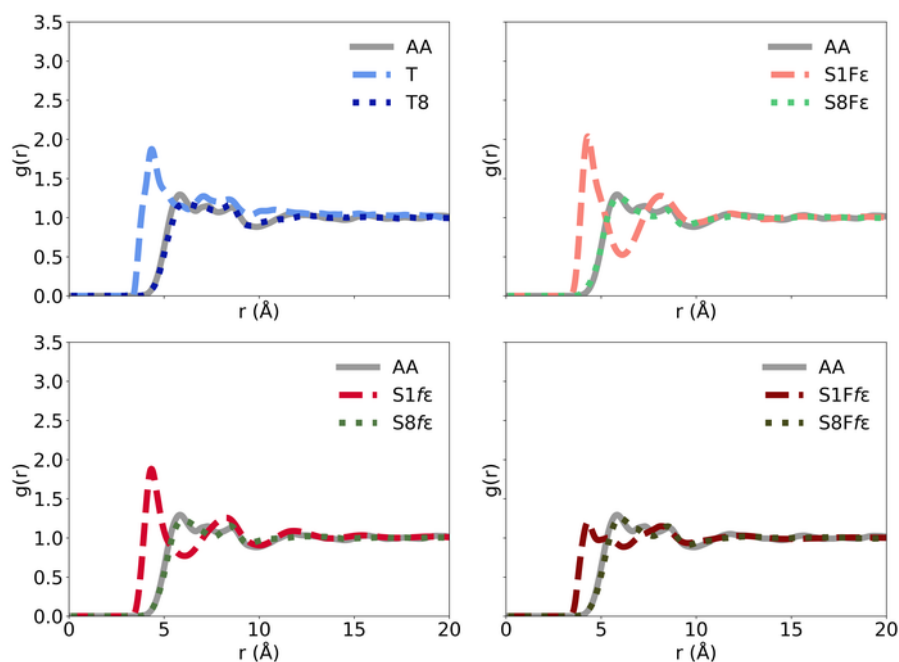


Figure S19: Urea-Urea RDFs, shown for all teacher and student models using different force targets.

S4 Three-body evaluation with varying force targets

S4.1 Summary ADF TAE and ADFs of teacher and student models with varying force targets

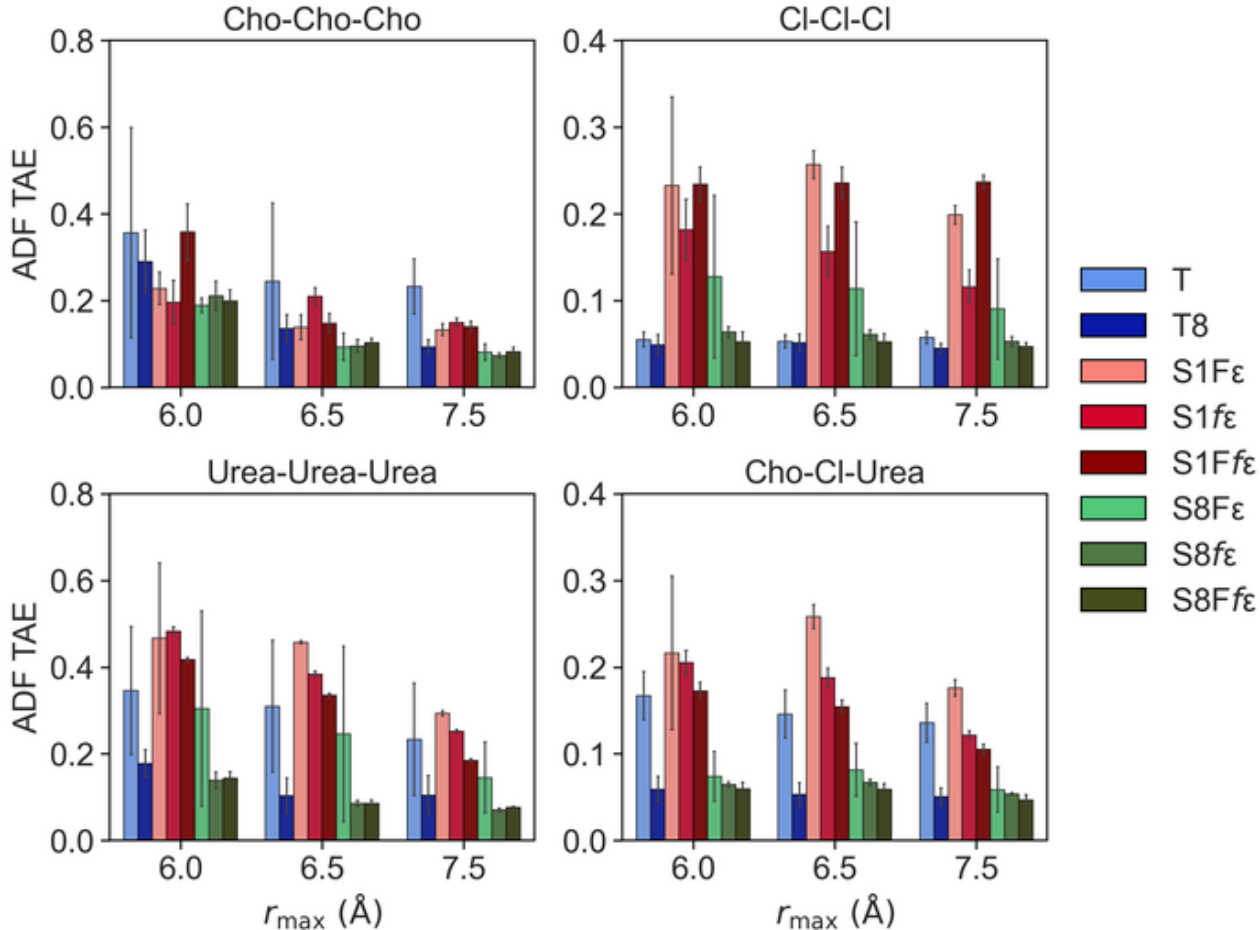


Figure S20: Comparison of ADF TAE at different ADF cutoff values $r_{\max}$ at for teacher and student models relative to the AA reference (using different force targets). Error bars denote one standard deviation over 8 replicas. Teacher (**T**) results were calculated as the mean of eight independent MD simulations, each performed with a teacher model trained from a unique random seed. **S1** student variants are trained from a single teacher and then run in eight independent replicas; their reported result is the average across all eight student models, with each model trained from one of the individual teachers. **S8** models are trained on averaged data from all eight teachers. Regarding energy data inclusion, student models here were trained to only per-atom energies. **F**: ground-truth forces, **f**: teacher forces, **E**: total energy, ϵ : per-bead energies.

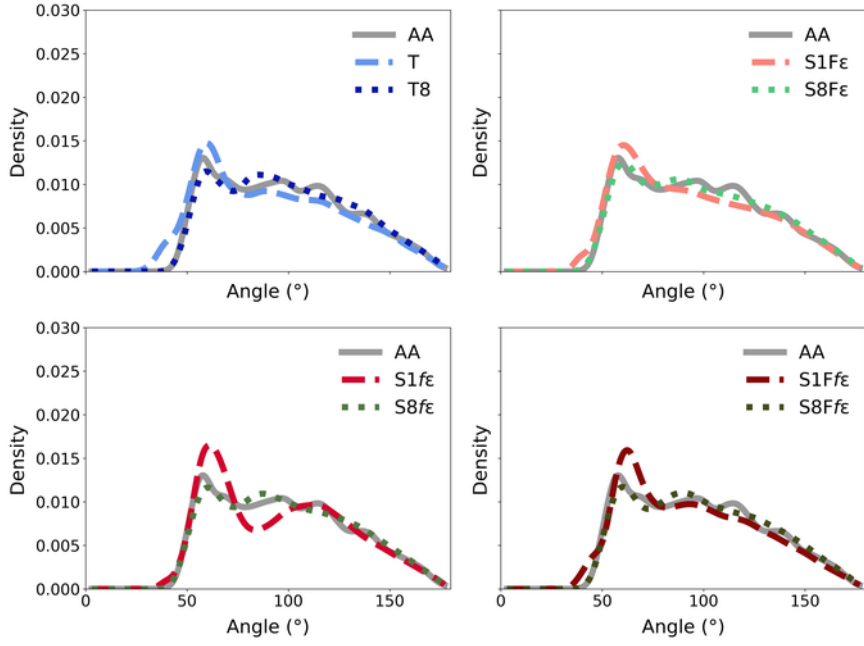


Figure S21: Cho-Cho-Cho ADF distribution using $r_{\max} = 7.5 \text{ \AA}$ shown for all teacher and student models using different force targets.

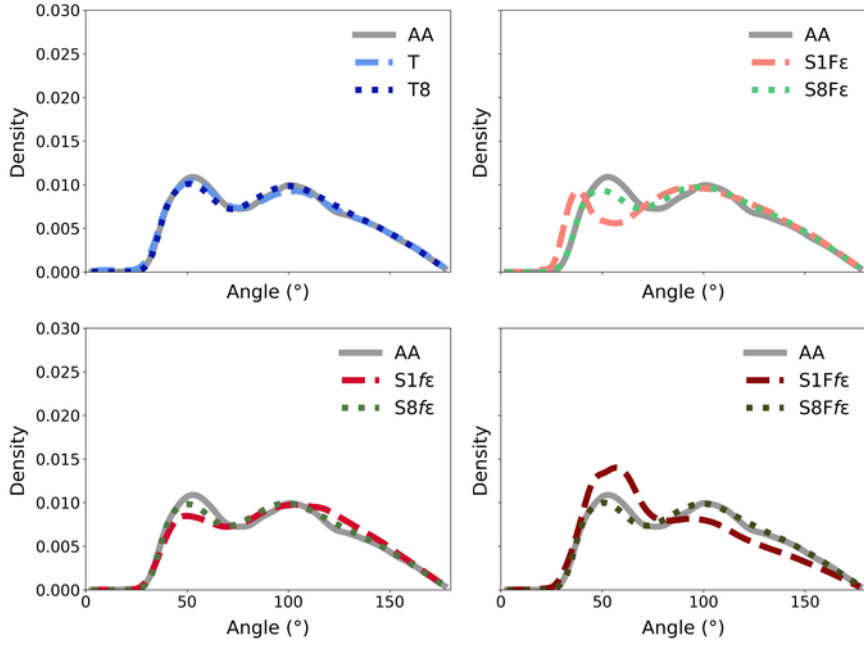


Figure S22: Cl-Cl-Cl ADF distribution using $r_{\max} = 7.5 \text{ \AA}$ shown for all teacher and student models using different force targets.

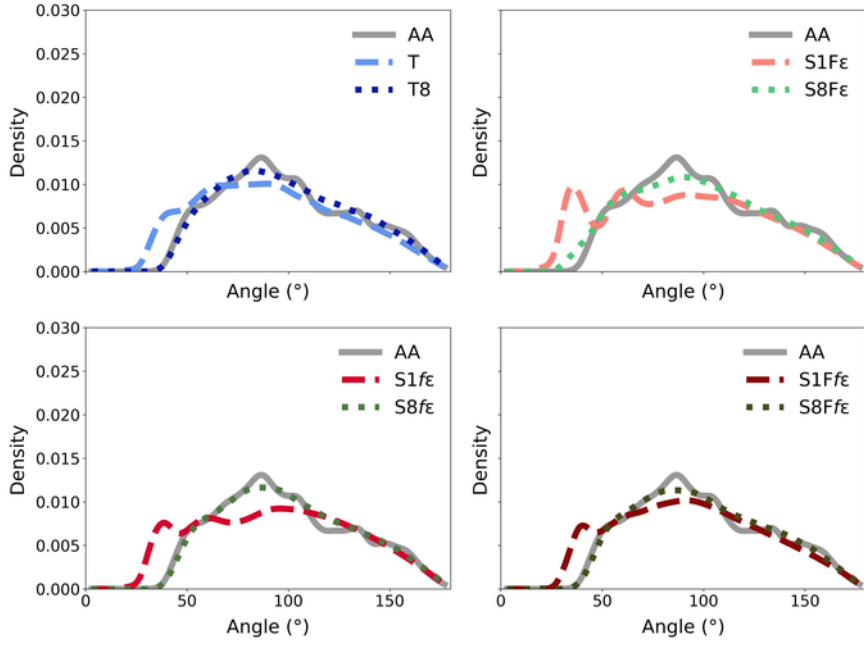


Figure S23: Urea-Urea-Urea ADF distribution using $r_{\max} = 7.5 \text{ \AA}$ shown for all teacher and student models using different force targets.

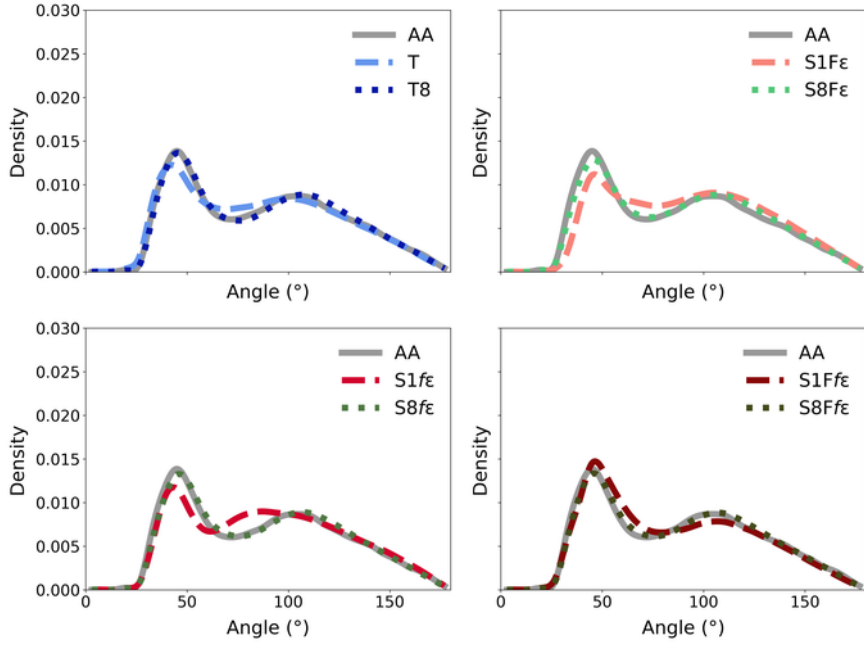


Figure S24: Cho-Cl-Urea ADF distribution using $r_{\max} = 7.5 \text{ \AA}$ shown for all teacher and student models using different force targets.

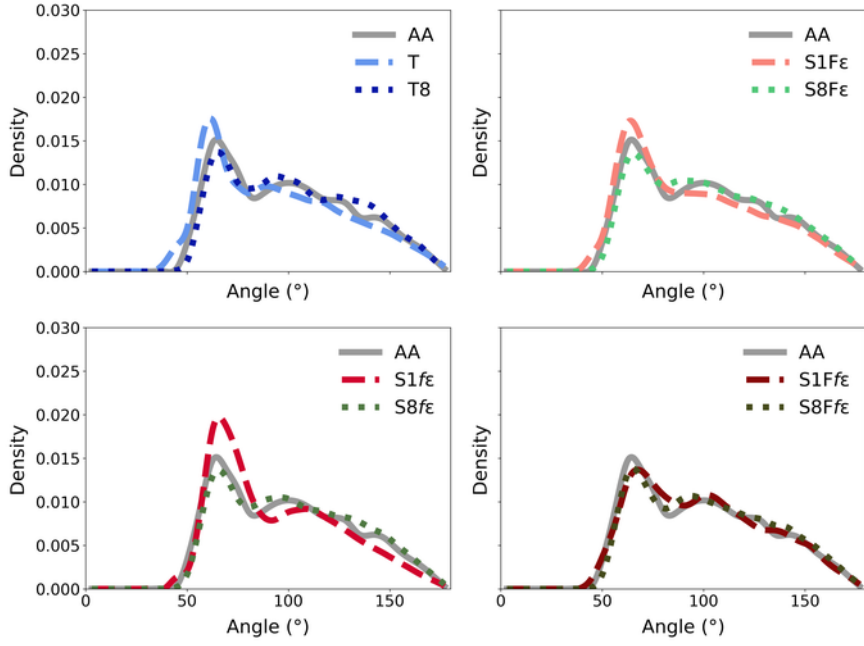


Figure S25: Cho-Cho-Cho ADF distribution using $r_{\max} = 6.5 \text{ \AA}$ shown for all teacher and student models using different force targets.

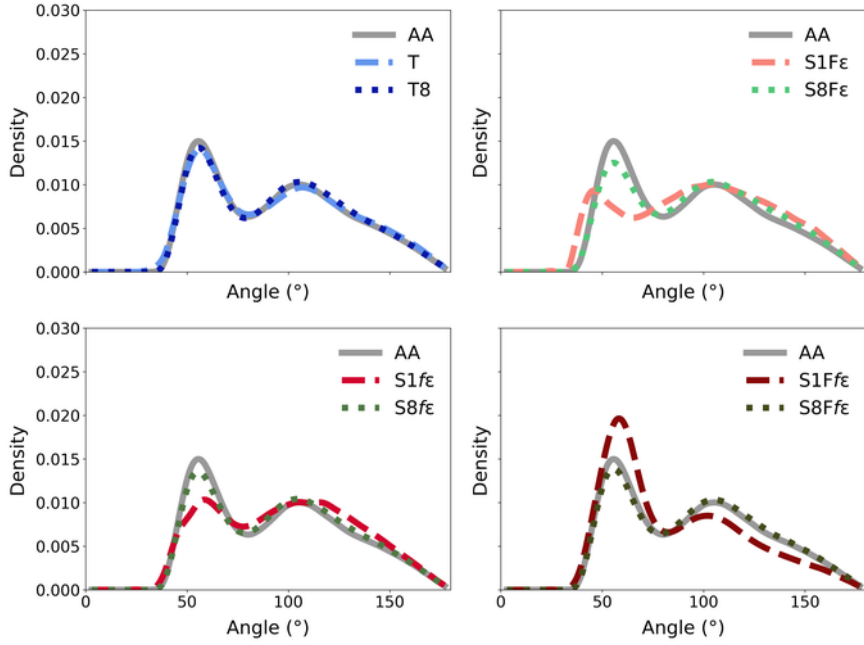


Figure S26: Cl-Cl-Cl ADF distribution using $r_{\max} = 6.5 \text{ \AA}$ shown for all teacher and student models using different force targets.

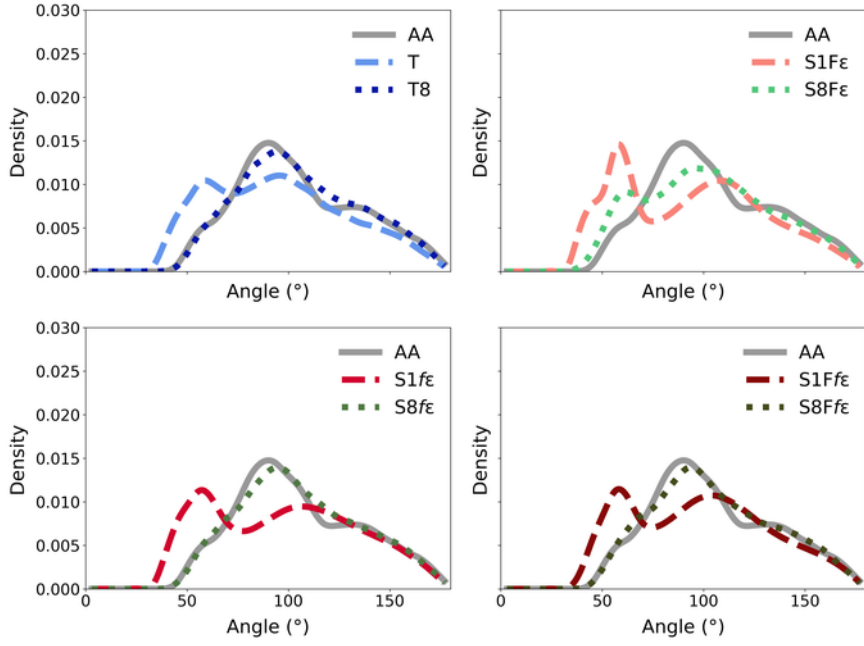


Figure S27: Urea-Urea-Urea ADF distribution using $r_{\max} = 6.5 \text{ \AA}$ shown for all teacher and student models using different force targets.

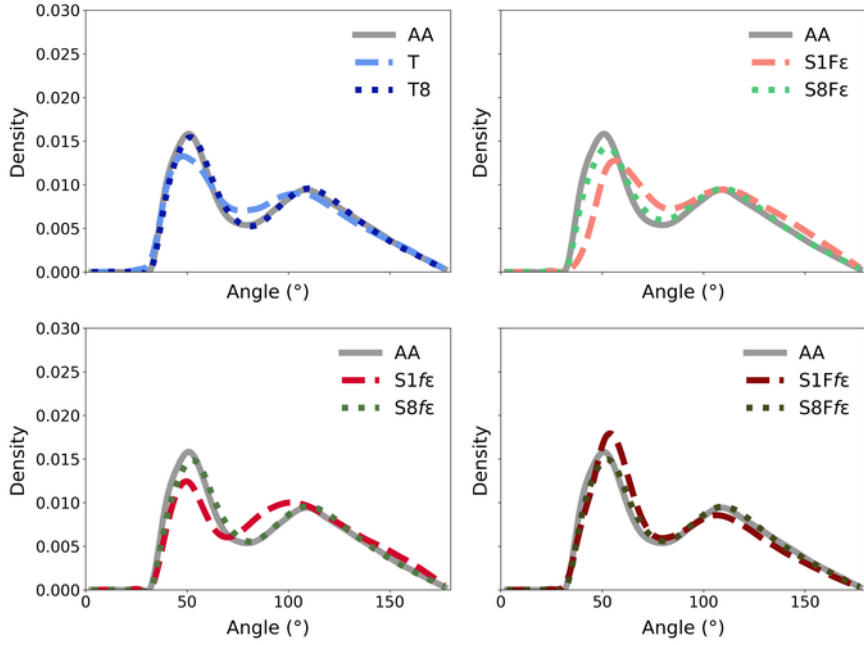


Figure S28: Cho-Cl-Urea ADF distribution using $r_{\max} = 6.5 \text{ \AA}$ shown for all teacher and student models using different force targets.

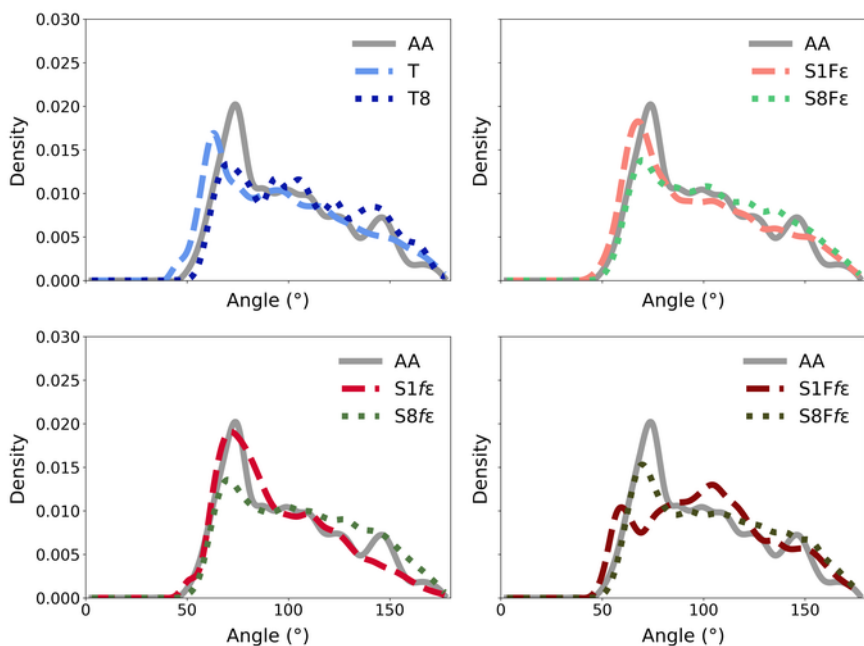


Figure S29: Cho-Cho-Cho ADF distribution using $r_{\max} = 6.0 \text{ \AA}$ shown for all teacher and student models using different force targets.

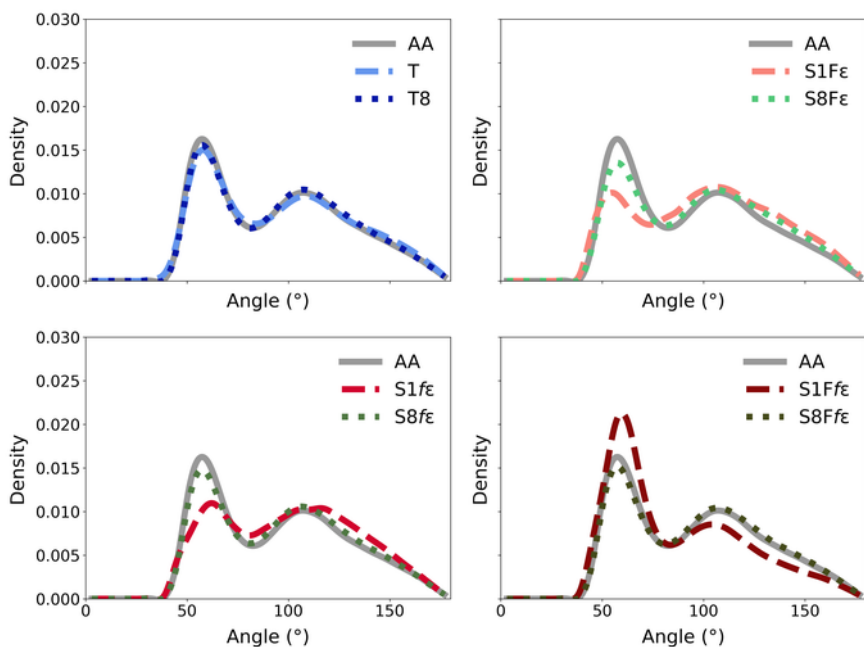


Figure S30: Cl-Cl-Cl ADF distribution using $r_{\max} = 6.0 \text{ \AA}$ shown for all teacher and student models using different force targets.

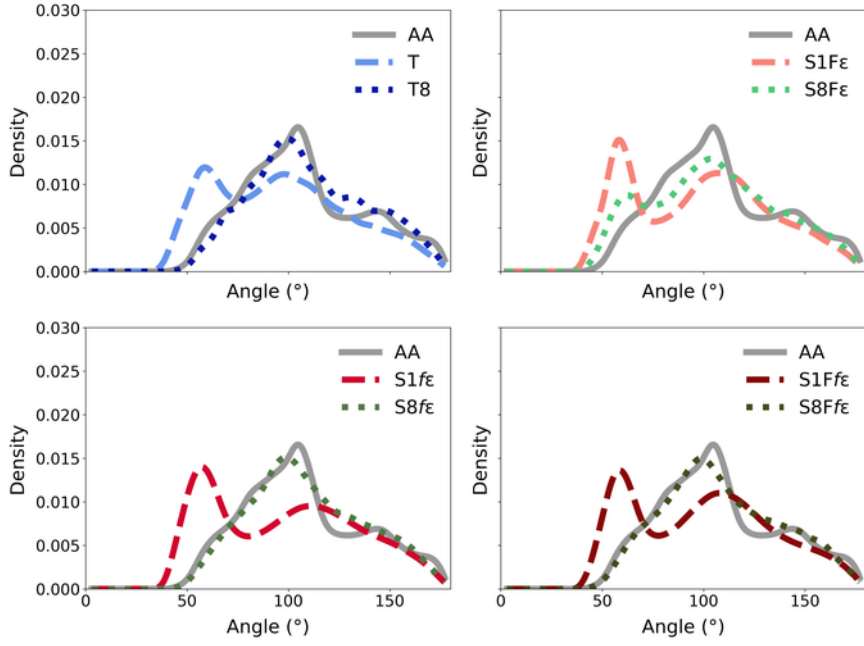


Figure S31: Urea-Urea-Urea ADF distribution using $r_{\max} = 6.0 \text{ \AA}$ shown for all teacher and student models using different force targets.

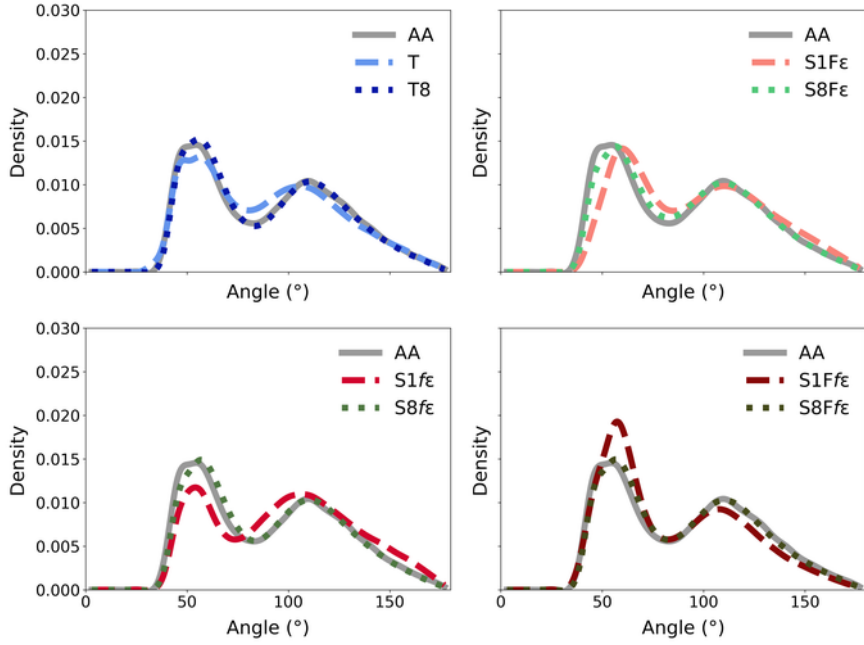


Figure S32: Cho-Cl-Urea ADF distribution using $r_{\max} = 6.0 \text{ \AA}$ shown for all teacher and student models using different force targets.

S5 Many-body evaluation with varying force targets

S5.1 CDF TAE and CDFs of teacher models

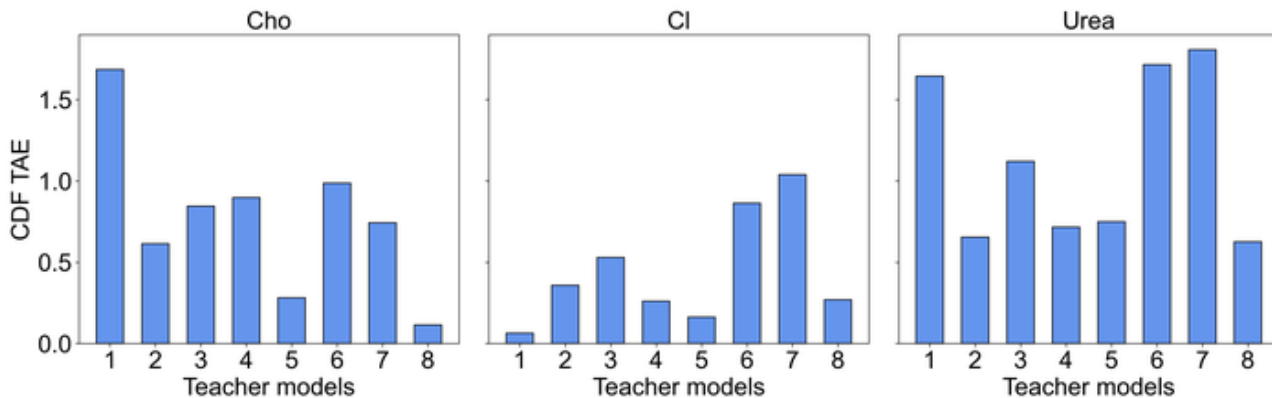


Figure S33: Comparison of CDF TAE of each species, shown for each teacher model trained on the same data and architecture but with different random seeds.

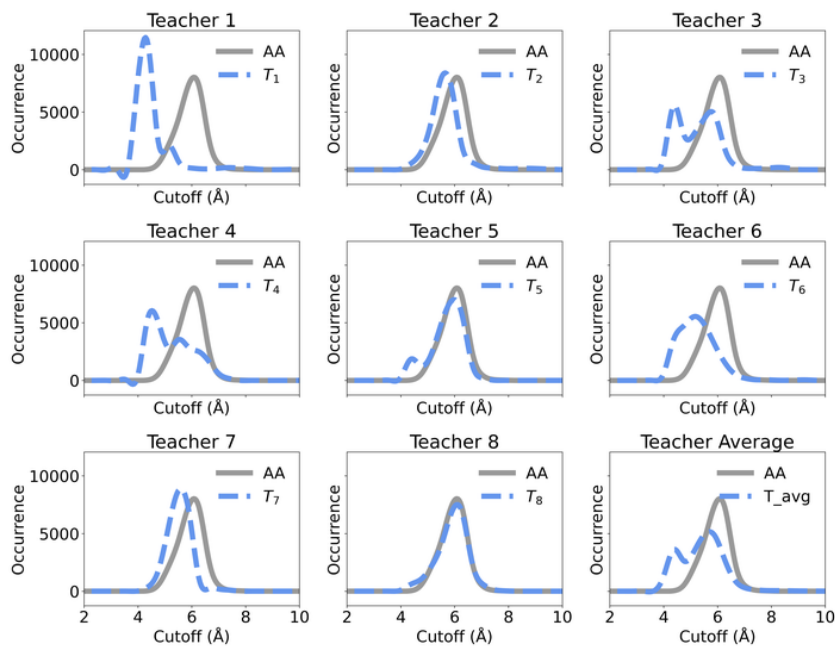


Figure S34: Choline CDF distribution, shown for each teacher model trained on the same data and architecture but with different random seeds.

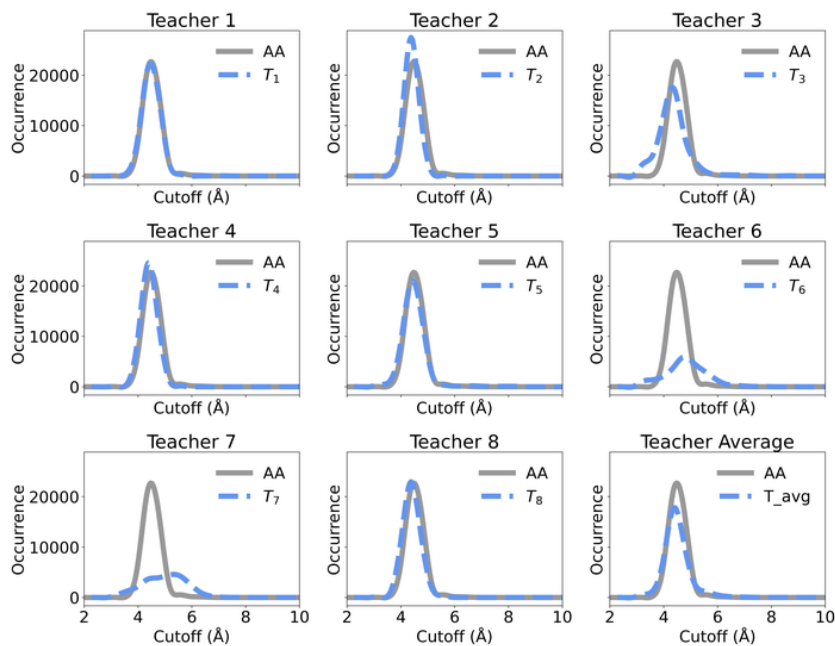


Figure S35: Chloride CDF distribution, shown for each teacher model trained on the same data and architecture but with different random seeds.

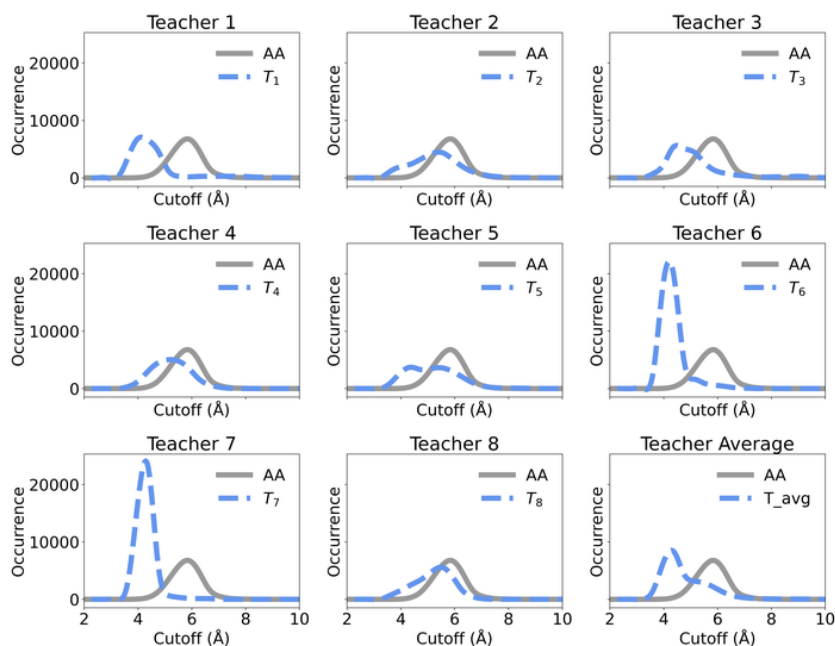


Figure S36: Urea CDF distribution, shown for each teacher model trained on the same data and architecture but with different random seeds.

S5.2 CDF TAE of single teacher-trained student models with varying force targets

Figure S37 shows the RDF TAEs for each of the eight **S1** models, each trained from its corresponding teacher. Figure S38 presents the overall RDF TAE aggregated across all pairs.

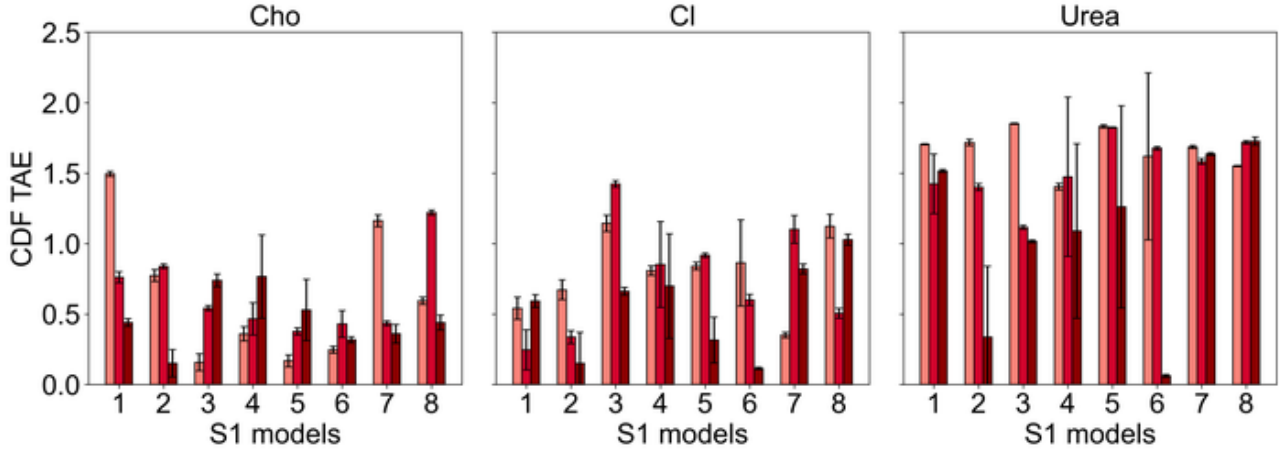


Figure S37: Comparison of CDF TAE for student models (using different force targets) relative to the AA reference. Each model was trained from respective individual teachers 1-8. Error bars denote one standard deviation over 8 replicas. Regarding energy data inclusion, the models were trained to only per-atom energies. F : ground-truth forces, f : teacher forces, E : total energy, ϵ : per-bead energies.

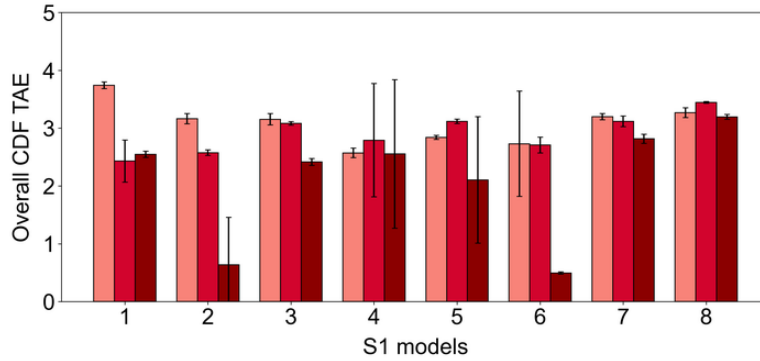


Figure S38: Comparison of overall CDF TAE for student models (using different force targets) relative to the AA reference. Each model was trained from respective individual teachers 1-8. For each model, the overall CDF TAE is calculated as the sum of the CDF TAEs of all (6) pairs in the system. Error bars denote one standard deviation over 8 replicas. Regarding energy data inclusion, the models were trained to only per-atom energies. F : ground-truth forces, f : teacher forces, E : total energy, ϵ : per-bead energies.

S5.3 Summary CDF TAE and CDFs of teacher and student models with varying force targets

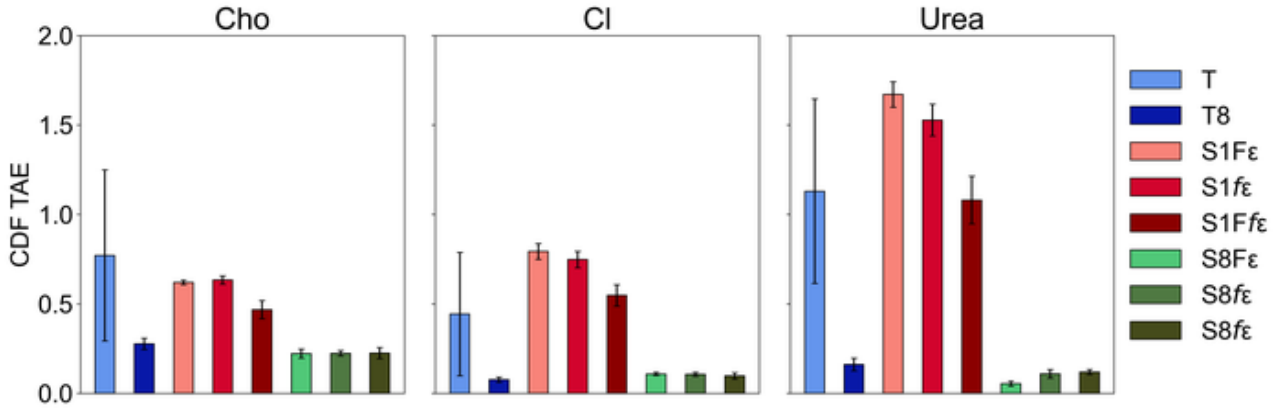


Figure S39: Comparison of CDF TAE for teacher and student models (using different force targets) relative to the AA reference. Error bars denote one standard deviation over 8 replicas. Teacher (**T**) results were calculated as the mean of eight independent MD simulations, each performed with a teacher model trained from a unique random seed. **S1** student variants are trained from a single teacher and then run in eight independent replicas; their reported result is the average across all eight student models, with each model trained from one of the individual teachers. **S8** models are trained on averaged data from all eight teachers. Regarding energy data inclusion, student models here were trained to only per-atom energies. **F**: ground-truth forces, **f**: teacher forces, **E**: total energy, ϵ : per-bead energies.

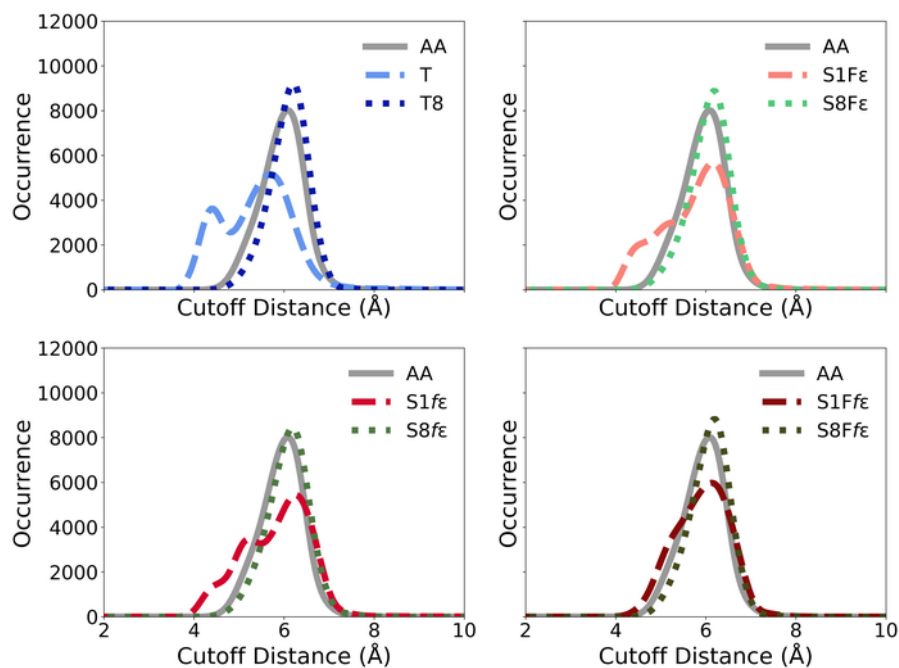


Figure S40: Choline CDF distribution, shown for all teacher and student models using different force targets.

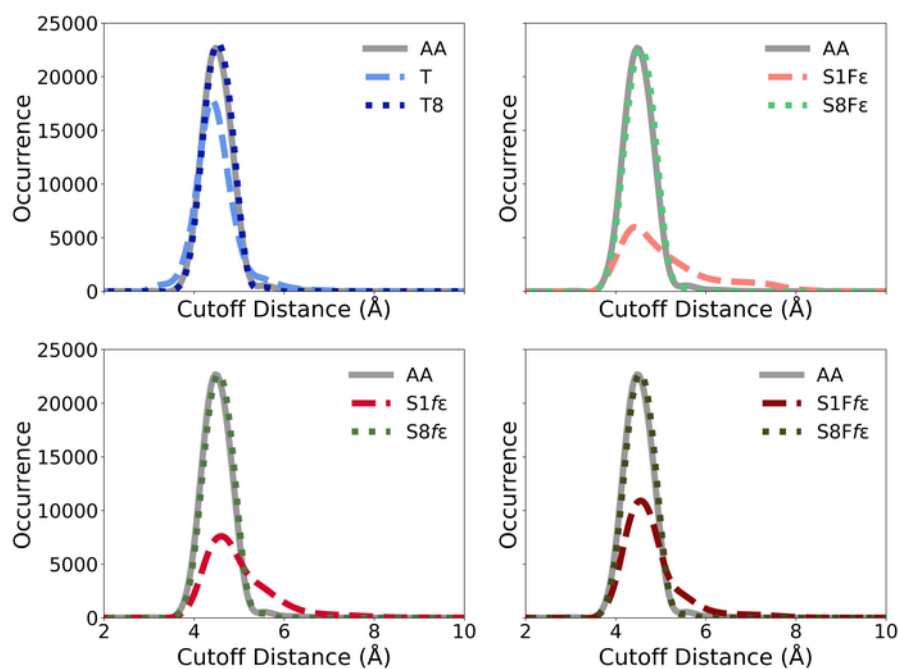


Figure S41: Chloride CDF distribution, shown for all teacher and student models using different force targets.

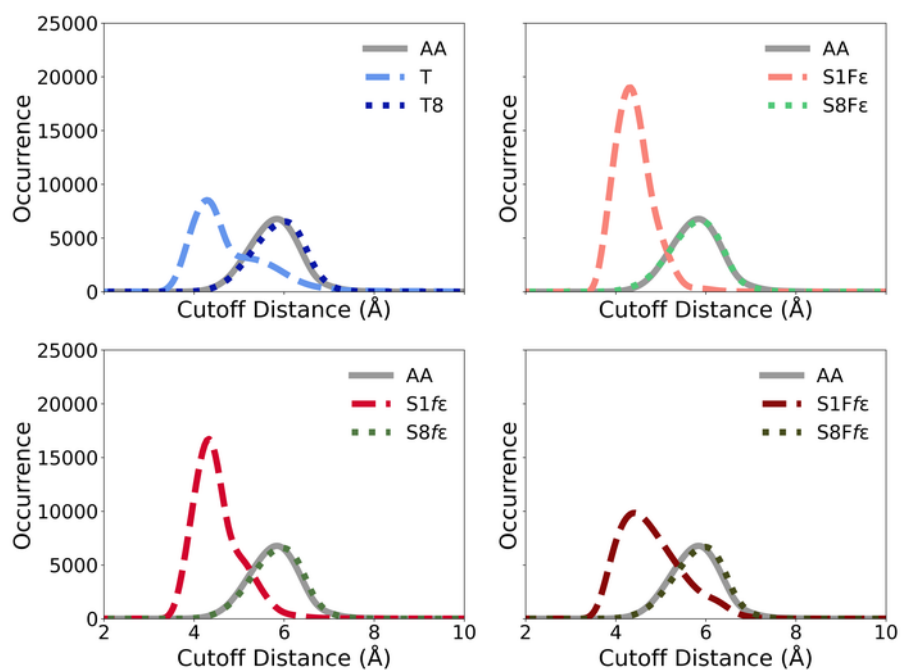


Figure S42: Urea CDF distribution, shown for all teacher and student models using different force targets.

S6 Two-body evaluation with varying energy targets

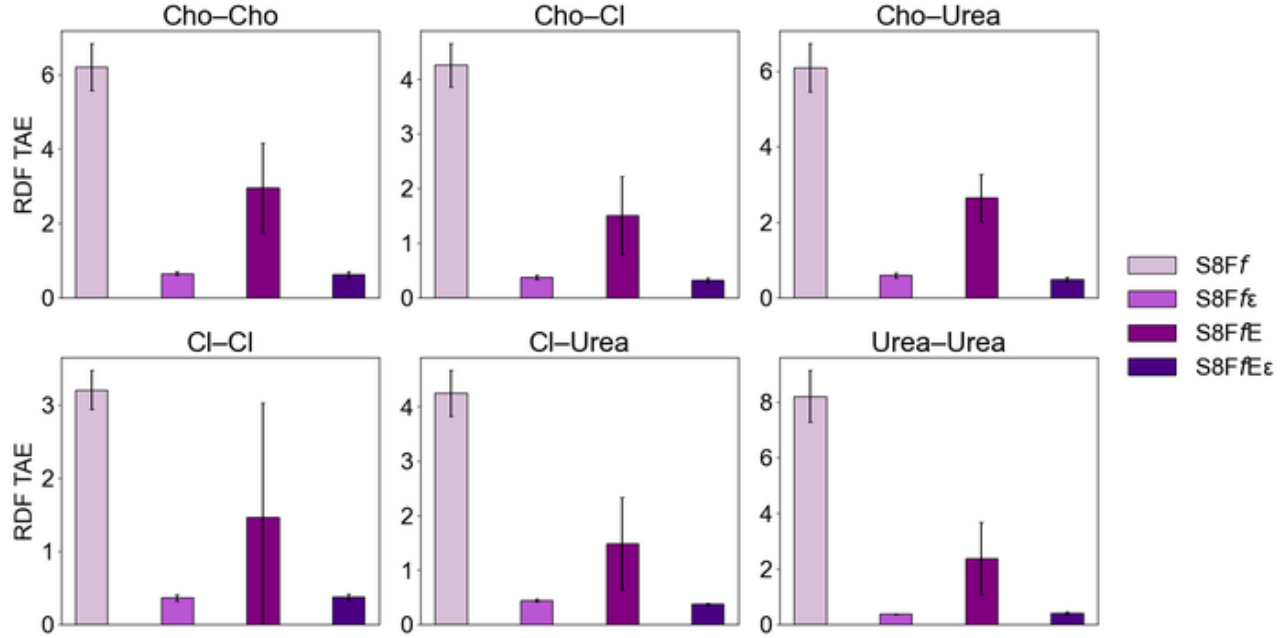


Figure S43: Comparison of RDF TAE for student models using different energy targets relative to the AA reference. Error bars denote one standard deviation over 8 replicas. **S8** models are trained on averaged data from all eight teachers. Regarding force data inclusion, the models were trained to both ground-truth and teacher forces. **F**: ground-truth forces, **f**: teacher forces, **E**: total energy, **ε**: per-bead energies.

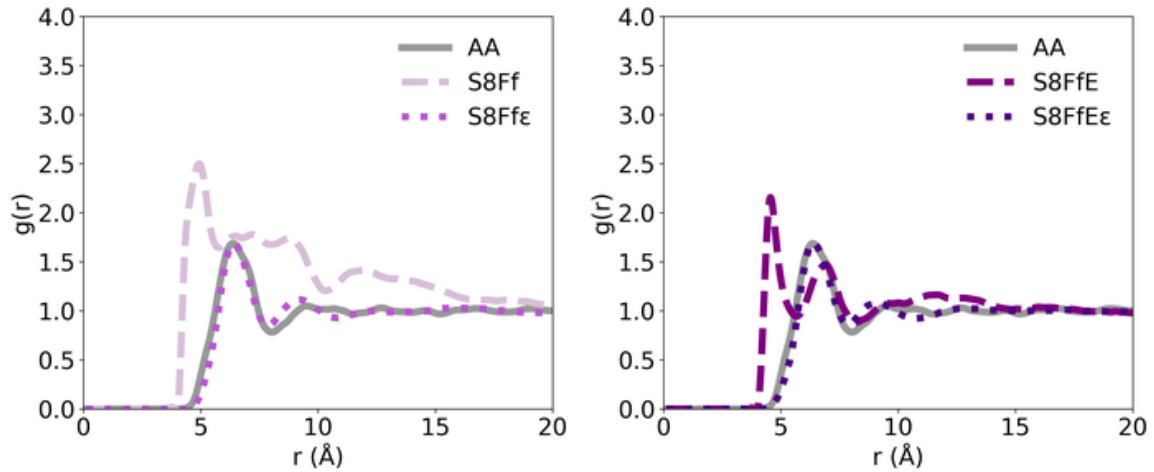


Figure S44: Cho-Cho RDFs, shown for the student models trained from an ensemble of teachers using different energy targets.

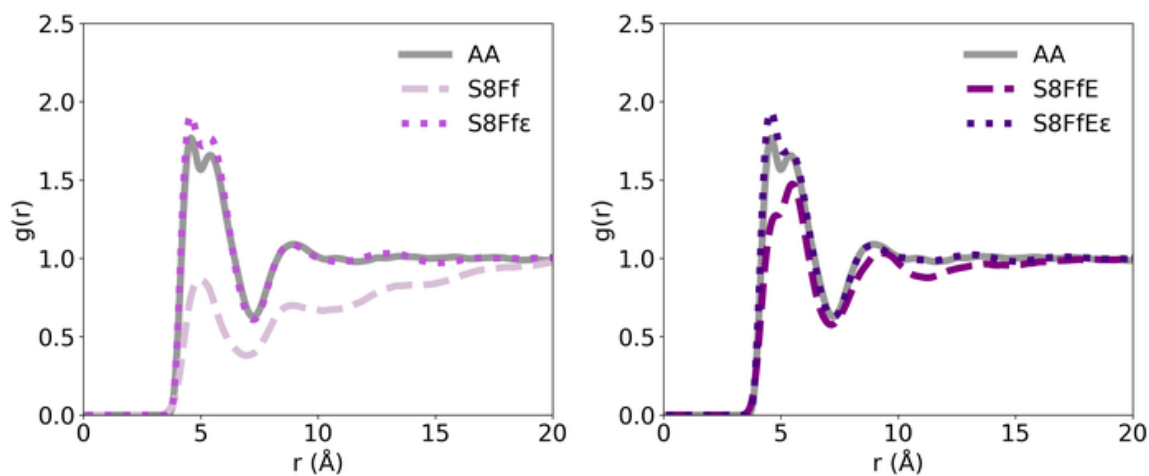


Figure S45: Cho-Cl RDFs, shown for the student models trained from an ensemble of teachers using different energy targets.

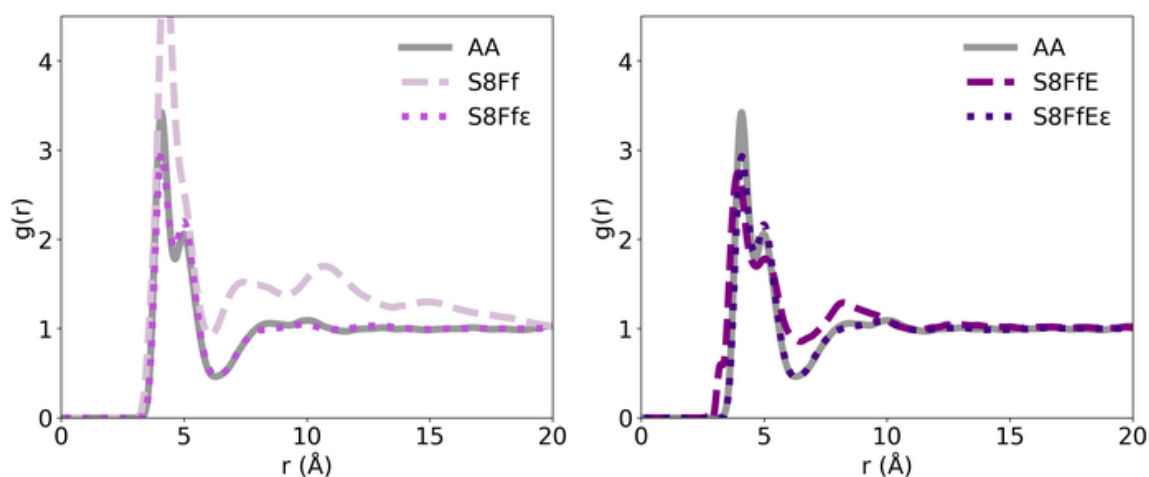


Figure S46: Cho-Urea RDFs, shown for the student models trained from an ensemble of teachers using different energy targets.

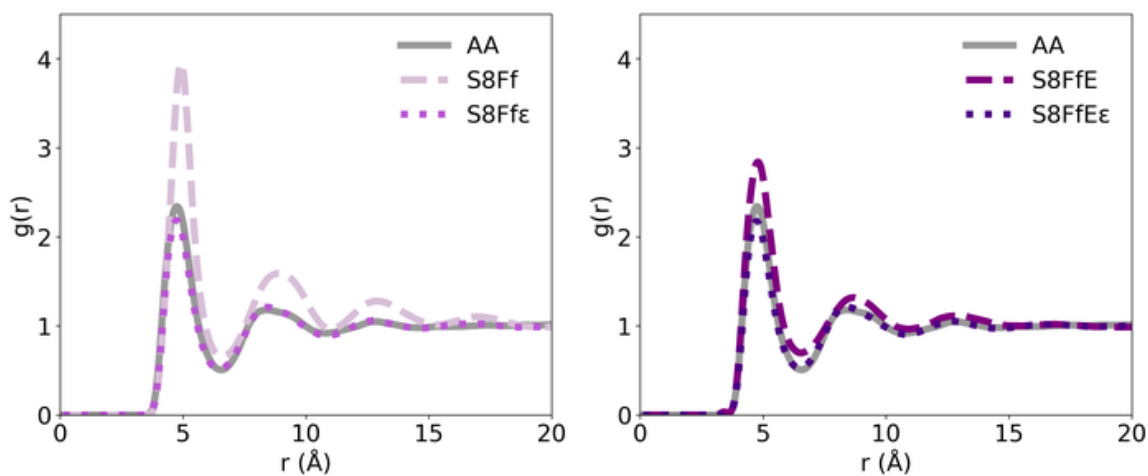


Figure S47: Cl-Cl RDFs, shown for the student models trained from an ensemble of teachers using different energy targets.

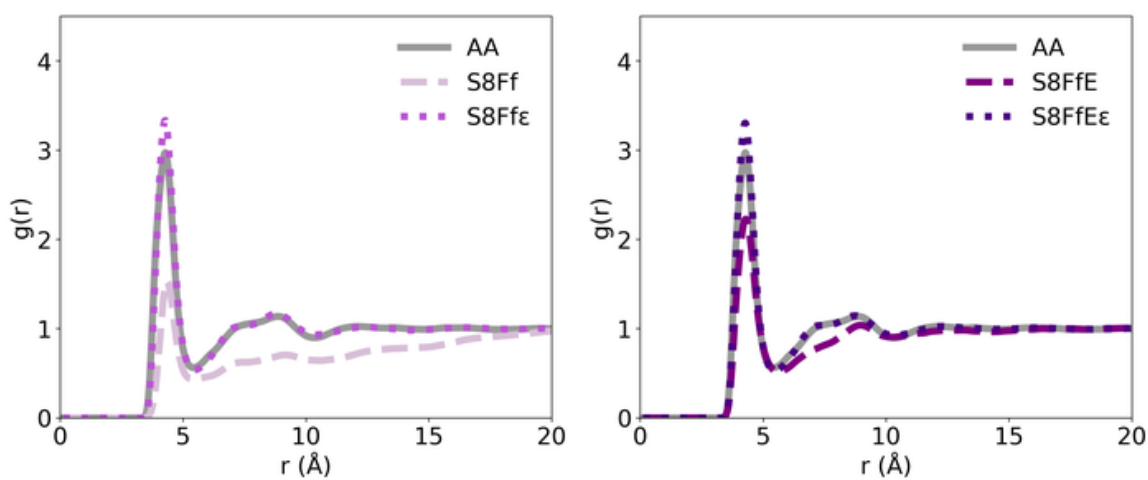


Figure S48: Cl-Urea RDFs, shown for the student models trained from an ensemble of teachers using different energy targets.

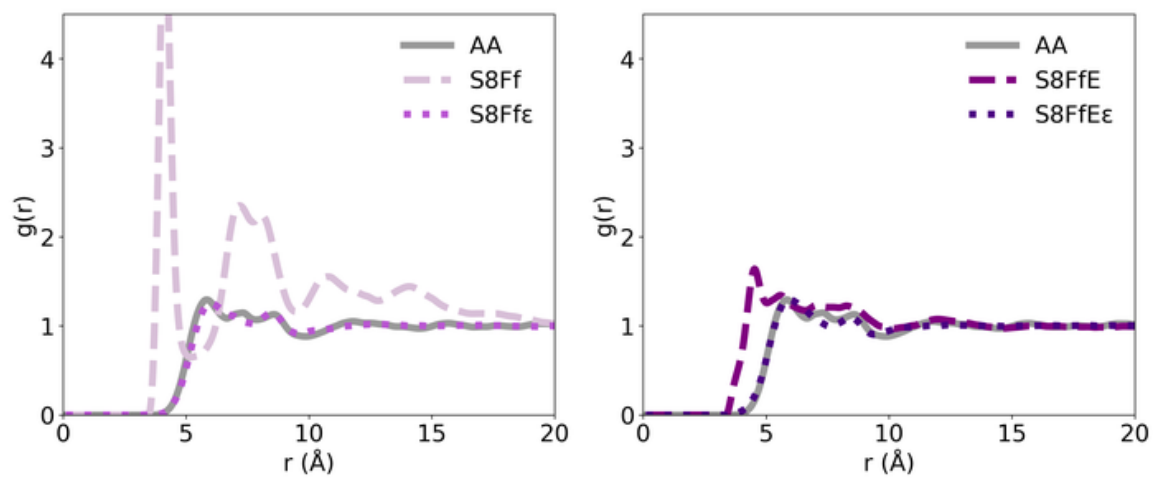


Figure S49: Urea-Urea RDFs, shown for the student models trained from an ensemble of teachers using different energy targets.

S7 Three-body evaluation with varying energy targets

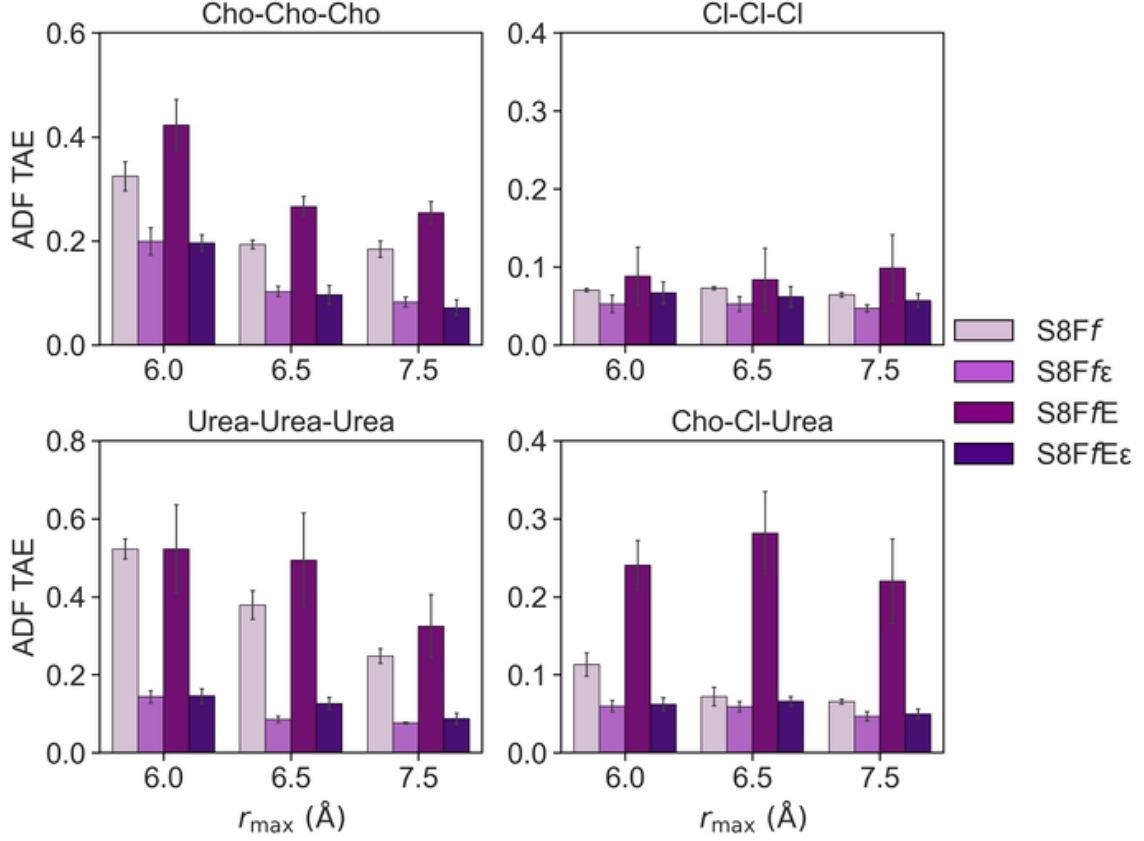


Figure S50: Comparison of ADF TAE at different ADF cutoff values $r_{\max}$ for teacher and student models relative to the AA reference, using different energy targets. Error bars denote one standard deviation over 8 replicas. Regarding force data inclusion, student models here were trained to both ground-truth and teacher forces. **F**: ground-truth forces, **f**: teacher forces, **E**: total energy, **ϵ** : per-bead energies.

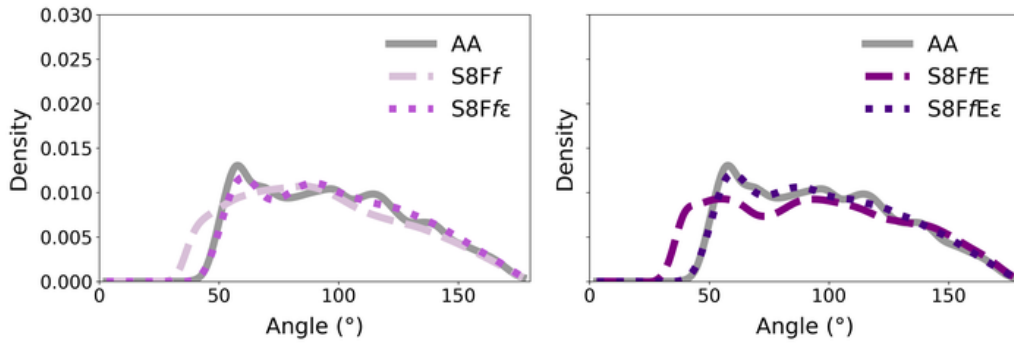


Figure S51: Cho-Cho-Cho ADF distribution using $r_{\max} = 7.5$ Å shown for all teacher and student models using different force targets.

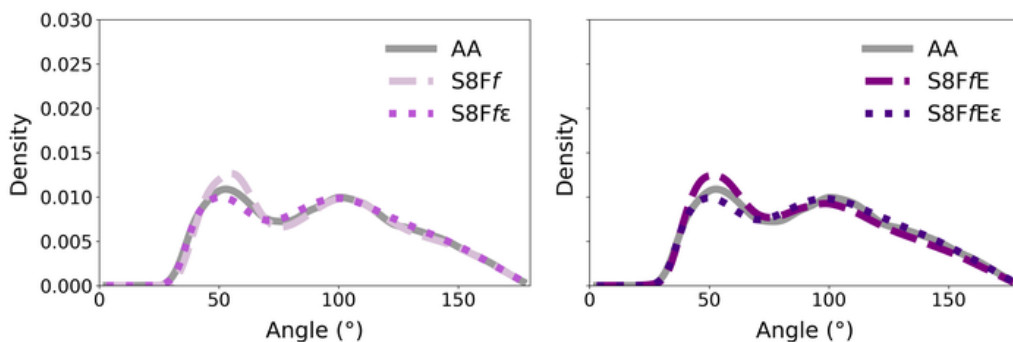


Figure S52: Cl-Cl-Cl ADF distribution using $r_{\max} = 7.5 \text{ \AA}$ shown for all teacher and student models using different force targets.

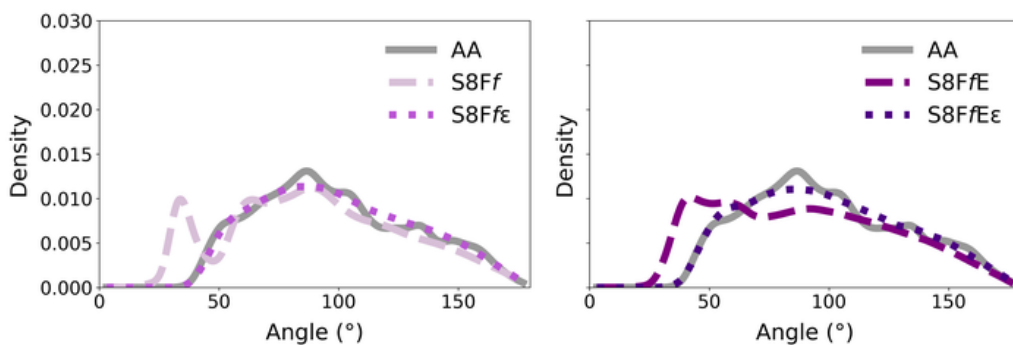


Figure S53: Urea-Urea-Urea ADF distribution using $r_{\max} = 7.5 \text{ \AA}$ shown for all teacher and student models using different force targets.

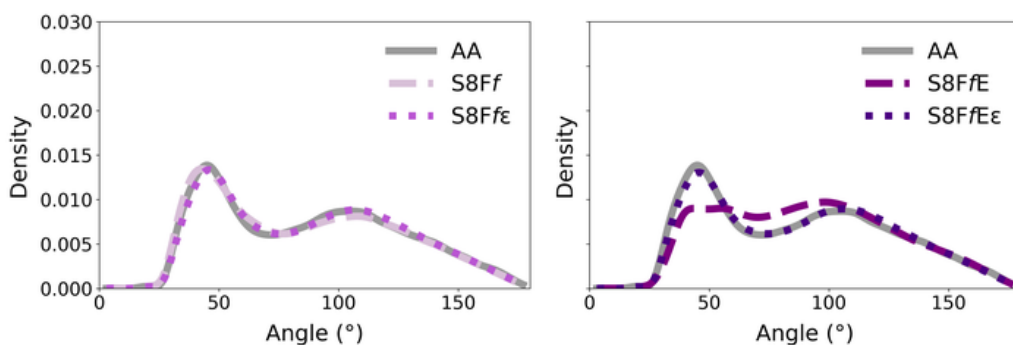


Figure S54: Cho-Cl-Urea ADF distribution using $r_{\max} = 7.5 \text{ \AA}$ shown for all teacher and student models using different force targets.

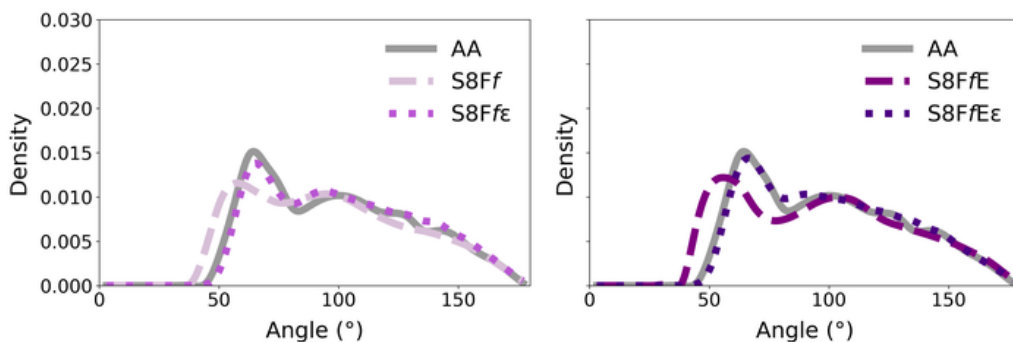


Figure S55: Cho-Cho-Cho ADF distribution using $r_{\max} = 6.5 \text{ \AA}$ shown for all teacher and student models using different force targets.

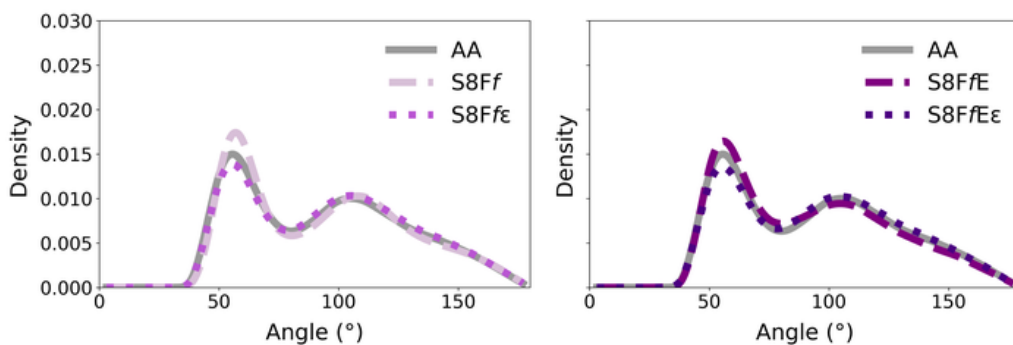


Figure S56: Cl-Cl-Cl ADF distribution using $r_{\max} = 6.5 \text{ \AA}$ shown for all teacher and student models using different force targets.

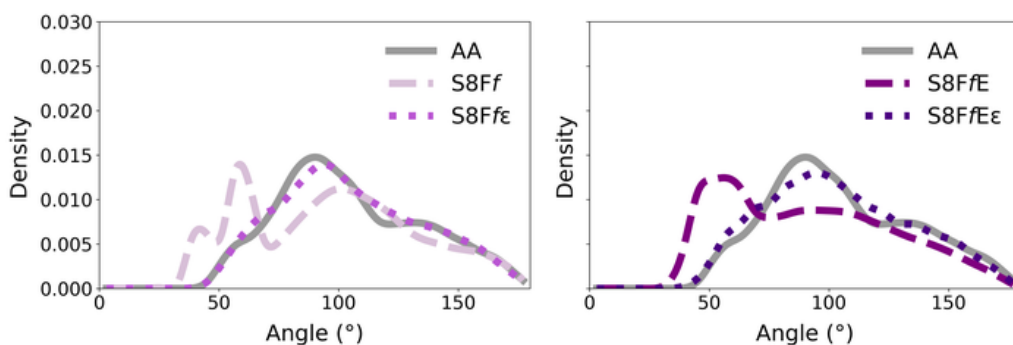


Figure S57: Urea-Urea-Urea ADF distribution using $r_{\max} = 6.5 \text{ \AA}$ shown for all teacher and student models using different force targets.

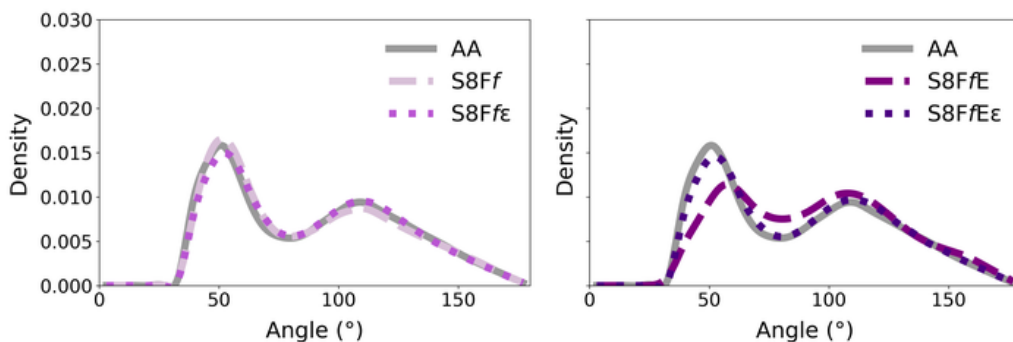


Figure S58: Cho-Cl-Urea ADF distribution using $r_{\max} = 6.5 \text{ \AA}$ shown for all teacher and student models using different force targets.

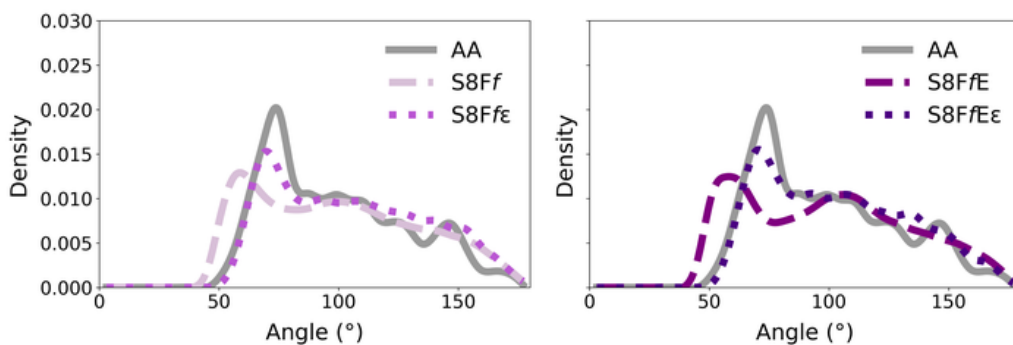


Figure S59: Cho-Cho-Cho ADF distribution using $r_{\max} = 6.0 \text{ \AA}$ shown for all teacher and student models using different force targets.

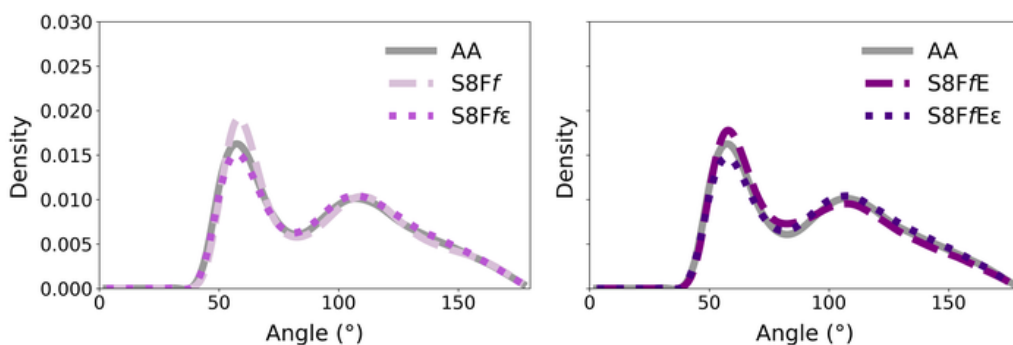


Figure S60: Cl-Cl-Cl ADF distribution using $r_{\max} = 6.0 \text{ \AA}$ shown for all teacher and student models using different force targets.

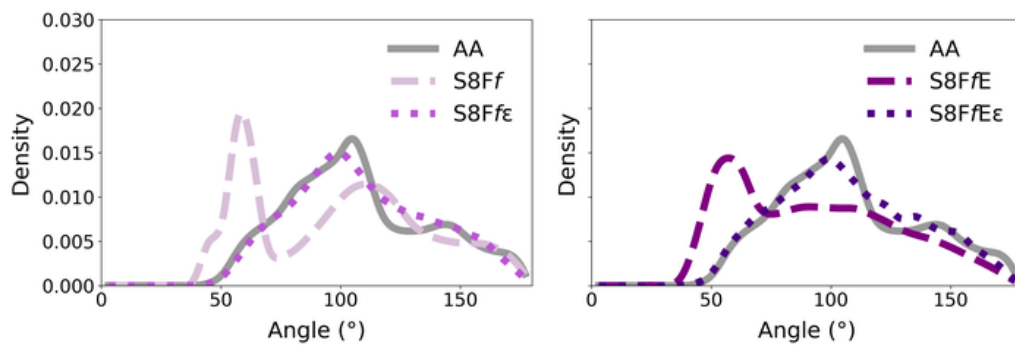


Figure S61: Urea-Urea-Urea ADF distribution using $r_{\max} = 6.0 \text{ \AA}$ shown for all teacher and student models using different force targets.

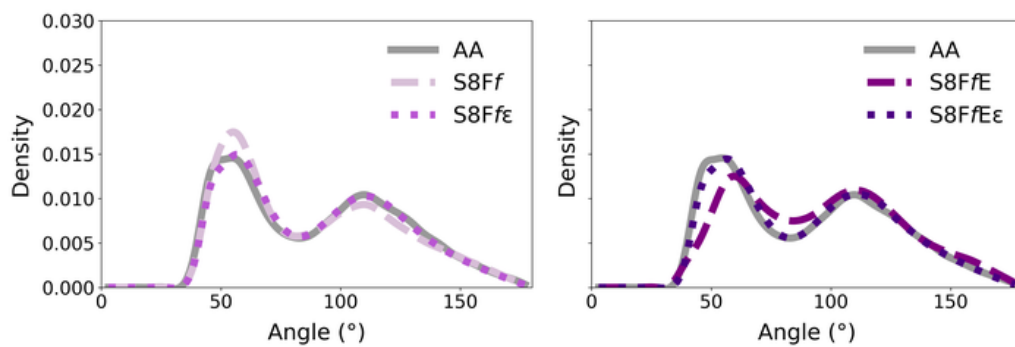


Figure S62: Cho-Cl-Urea ADF distribution using $r_{\max} = 6.0 \text{ \AA}$ shown for all teacher and student models using different force targets.

S8 Many-body evaluation with varying energy targets

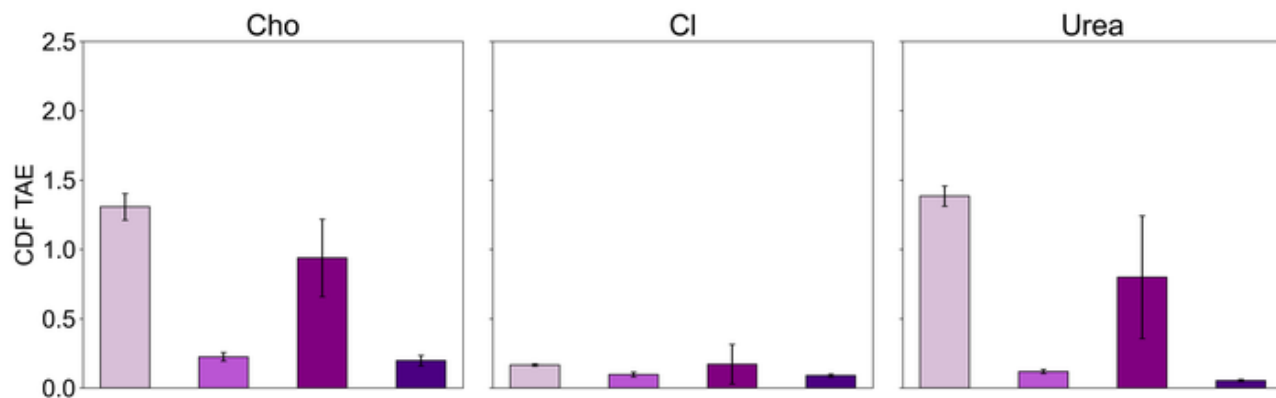


Figure S63: Comparison of CDF TAE for student models using different energy targets relative to the AA reference.

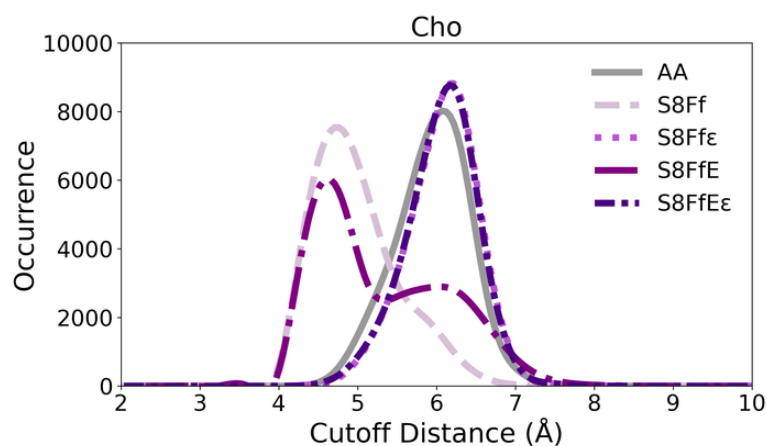


Figure S64: Choline CDF distribution, shown for the student models trained from an ensemble of teachers using different energy targets.

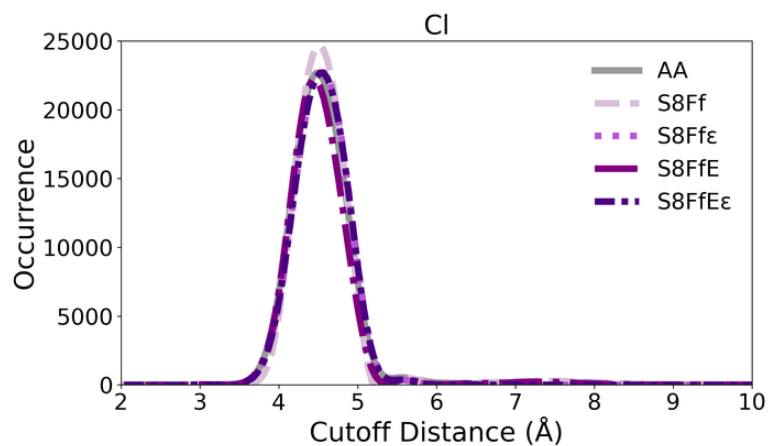


Figure S65: Chloride CDF distribution, shown for the student models trained from an ensemble of teachers using different energy targets.

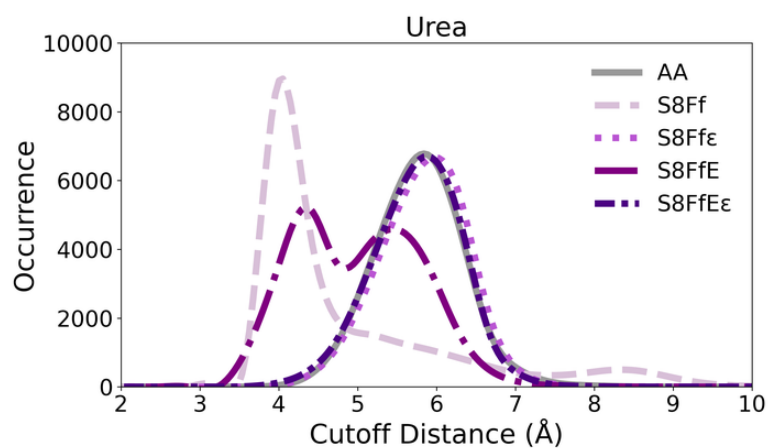


Figure S66: Urea CDF distribution, shown for the student models trained from an ensemble of teachers using different energy targets.

S9 Simulation Input Files

All-atom Molecular Dynamics simulations were performed using GROMACS, and the resulting trajectories were coarse-grained to generate the initial training dataset. The input files below (`packmol.in`, `topol.top`, `min.mdp`, `equil.mdp`, and `prod.mdp`) define the simulation parameters, topology, and system composition in this study. The `.mdp` files were adapted from `grompp-prod.mdp` from the CCU-Tutorial in the OPLS-DES repository <https://github.com/orlandoacevedo/DES>.

1. **System Preparation:** The individual component structures `choline.pdb`, `chloride.pdb`, and `urea.pdb` were obtained from the OPLS-DES repository <https://github.com/orlandoacevedo/DES>.
2. **System Construction:** Initial configurations were constructed using Packmol:

```
packmol < packmol.in
```

3. **Energy Minimization and Equilibration:** The following commands were employed to perform energy minimization and system equilibration:

```
gmxd grompp -f min.mdp -c system.pdb -p topol.top -o em.tpr
gmxd mdrun -deffnm em
gmxd grompp -f equil.mdp -c em.gro -p topol.top -o eq.tpr
gmxd mdrun -deffnm eq -v
```

4. **Production Run:** The commands below were used to produce the trajectory data:

```
gmxd grompp -f prod.mdp -c eq.gro -p topol.top -o prod.tpr
gmxd mdrun -deffnm prod -v
```

5. **Trajectory Processing:** The AA trajectories were post-processed and transformed to coarse-grained representations, which served as initial training data for our models.

`topol.top`

```
#define _FF_OPLS
#define _FF_OPLSAA

[defaults]
; nbfunc      comb-rule    gen-pairs    fudgeLJ    fudgeQQ
1              3           yes          0.5        0.5

; === LOAD ATOM TYPES ===
```

```

#include "choline_atomtypes_DES.itp"
#include "cl_atomtypes_DES.itp"
#include "urea_atomtypes_DES.itp"

; === LOAD MOLECULES (*.itp) ===
#include "choline_DES.itp"
#include "cl_DES.itp"
#include "urea_DES.itp"

[system]
; Name
Neat CHOL Cl UREA

[molecules]
CHOL 250
Cl 250
UREA 500

```

packmol.in

```

seed 112233
tolerance 1.5
output system.pdb
filetype pdb
pbc 44.84 44.84 44.84

structure choline.pdb
  number 250
  inside cube 0. 0. 0. 44.84
end structure

structure chloride.pdb
  number 250
  inside cube 0. 0. 0. 44.84
end structure

structure urea.pdb
  number 500
  inside cube 0. 0. 0. 44.84
end structure

```

min.mdp

```

; RUN CONTROL PARAMETERS =
integrator = steep          ; Steepest descents minimization
nsteps = 50000              ; number of steps to integrate
emtol = 1000.0              ; Energy tolerance
emstep = 0.01               ; Step size for minimization

; NEIGHBORSEARCHING PARAMETERS =
cutoff-scheme = verlet     ; explicit cut-off at rvdw = rcoulomb
nstlist = 20                ; [steps] freq to update neighbor list
ns_type = grid              ; method of updating neighbor list

```

```

pbs = xyz                ; periodic boundary conditions in all directions
rlist = 1.6              ; [nm] cut-off distance for short-range neighbor
    list
verlet-buffer-tolerance = 0.005 ; max allowed error for pair interactions
    per particle

; OPTIONS FOR ELECTROSTATICS AND VDW =
coulombtype = PME        ; Particle-Mesh Ewald electrostatics
rcoulomb = 1.6           ; [nm] distance for Coulomb cut-off
vdw_type = PME           ; twin-range cut-off where rvdw >= rlist
rvdw = 1.6               ; [nm] distance for LJ cut-off
fourierspacing = 0.15    ; [nm] FFT grid spacing when using PME
pme_order = 4             ; interpolation order for PME, 4 = cubic
ewald_rtol = 1e-05        ; relative strength of Ewald-shifted potential at
    rcoulomb

; GENERATE VELOCITIES FOR STARTUP RUN =
gen_vel = yes            ; velocity generation turned on/off

; OPTIONS FOR BONDS =
constraints = hbonds
constraint_algorithm = lincs
continuation = no
lincs_order = 4
lincs_warnangle = 30
morse = no
lincs_iter = 2

```

equil.mdp

```

; RUN CONTROL PARAMETERS =
integrator = md           ; md integrator
tinit = 0                ; [ps] starting time for run
dt = 0.001               ; [ps] time step for integration
nsteps = 1000000         ; maximum number of steps to integrate
comm-mode = Linear        ; Remove center of mass translation

; NEIGHBORSEARCHING PARAMETERS =
cutoff-scheme = verlet    ; explicit, exact cut-off at rvdw = rcoulomb
nstlist = 20              ; [steps] freq to update neighbor list
ns_type = grid            ; method of updating neighbor list
pbs = xyz                ; periodic boundary conditions in all directions
rlist = 1.6              ; [nm] cut-off distance for short-range neighbor
    list
verlet-buffer-tolerance = 0.005 ; max allowed error for pair interactions
    per particle

; OPTIONS FOR ELECTROSTATICS AND VDW =
coulombtype = PME        ; Particle-Mesh Ewald electrostatics
rcoulomb = 1.6           ; [nm] distance for Coulomb cut-off
vdw_type = PME           ; twin-range cut-off where rvdw >= rlist
rvdw = 1.6               ; [nm] distance for LJ cut-off
fourierspacing = 0.15    ; [nm] FFT grid spacing when using PME

```

```

pme_order = 4           ; interpolation order for PME, 4 = cubic
ewald_rtol = 1e-05      ; relative strength of Ewald-shifted potential at
    rcoulomb

; OPTIONS FOR WEAK COUPLING ALGORITHMS =
tcoupl = v-rescale      ; temperature coupling method
tc-grps = System        ; groups to couple separately to temperature bath
tau_t = 1.0             ; [ps] time constant for coupling
ref_t = 298.15          ; reference temperature for coupling
pcoupl = no

; GENERATE VELOCITIES FOR STARTUP RUN =
gen_vel = no            ; velocity generation turned on/off

; OPTIONS FOR BONDS =
constraints = hbonds
constraint_algorithm = lincs
continuation = yes
lincs_order = 4
lincs_warnangle = 30
morse = no
lincs_iter = 2

```

prod.mdp

```

; RUN CONTROL PARAMETERS =
integrator = md          ; md integrator
tinit = 0               ; [ps] starting time for run
dt = 0.001              ; [ps] time step for integration
nsteps = 100000000       ; maximum number of steps to integrate
comm-mode = Linear       ; Remove center of mass translation

; OUTPUT CONTROL OPTIONS =
nstxout = 1000           ; [steps] freq to write coordinates to
    trajectory
nstvout = 1000           ; [steps] freq to write velocities to trajectory
nstfout = 1000           ; [steps] freq to write forces to trajectory
nstlog = 1000            ; [steps] freq to write energies to log file
nstenergy = 1000         ; [steps] freq to write to energy file
nstxout-compressed = 1000 ; freq to write coordinates to xtc trajectory

; NEIGHBORSEARCHING PARAMETERS =
cutoff-scheme = verlet   ; explicit, exact cut-off at rvdw = rcoulomb
nstlist = 20             ; [steps] freq to update neighbor list
ns_type = grid           ; method of updating neighbor list
pbc = xyz                ; periodic boundary conditions in all directions
rlist = 1.6              ; [nm] cut-off distance for short-range neighbor
    list
verlet-buffer-tolerance = 0.005 ; max allowed error for pair interactions
    per particle

; OPTIONS FOR ELECTROSTATICS AND VDW =
coulombtype = PME        ; Particle-Mesh Ewald electrostatics

```

```

rcoulomb = 1.6                ; [nm] distance for Coulomb cut-off
vdw_type = PME                ; twin-range cut-off where rvdw >= rlist
rvdw = 1.6                    ; [nm] distance for LJ cut-off
fourierspacing = 0.15        ; [nm] FFT grid spacing when using PME
pme_order = 4                  ; interpolation order for PME, 4 = cubic
ewald_rtol = 1e-05            ; relative strength of Ewald-shifted potential
    at rcoulomb

; OPTIONS FOR WEAK COUPLING ALGORITHMS =
tcoupl = v-rescale             ; temperature coupling method
tc-grps = System               ; groups to couple separately to temperature
    bath
tau_t = 1.0                    ; [ps] time constant for coupling
ref_t = 298.15                 ; reference temperature for coupling
pcoupl = no

; GENERATE VELOCITIES FOR STARTUP RUN =
gen_vel = no                   ; velocity generation turned on/off

; OPTIONS FOR BONDS =
constraints = hbonds
constraint_algorithm = lincs
continuation = yes
lincs_order = 4
lincs_warnangle = 30
morse = no
lincs_iter = 2

```

Coarse-grained Molecular Dynamics simulations were performed using LAMMPS. The input script (`in.lammps`) defines the simulation setup used for both the teacher and student models (e.g., `mliap_unified_hippynn.pt`). The training scripts employed to develop these models are available in the hippynn repository: <https://github.com/lanl/hippynn>.

in.lammps

```

units                real
atom_style            atomic
boundary              p p p

package              omp 8
suffix               omp

read_data             system.data

# Provide LAMMPS access to the repulsive potential defined externally
python source here """
import sys
import os
sys.path.insert(0, ".../myhippynn")
print("PYTHONPATH inside LAMMPS:", sys.path)

sys.path.append('.') # Add parent directory to path

```

```

"""

pair_style          mliap unified ../model/mliap_unified_hippynn.pt 0
pair_coeff          * * A B C      # same order of species names used during
                        training

neighbor            2.0 bin
neigh_modify        delay 0 every 1 check yes

timestep            2      # fs

thermo_style        custom step vol temp press ke pe etotal
thermo              100
thermo_modify       flush yes

min_style           cg
minimize            1.0e-6 1.0e-8 1000 10000

velocity            all create 298.15 91154341 mom yes rot yes dist gaussian

restart             1000 restart.*

fix                 NVT all nvt/omp temp 298.15 298.15 100.0

# Equilibration
run                 5000

# Production
dump                trj all custom 100 dump.out id type x y z xu yu zu vx vy
                        vz fx fy fz
dump_modify         trj sort id
dump                8 all xyz 100 traj.xyz
thermo              100

restart             1000 restart.*

run                 1000000

```