

# Solvation Structure Around Nanoparticles

## Integrating MD Simulations and Machine Learning

Shuvam Banerji Seal (22MS076)

October 25, 2025

# Outline I

- 1 Why the Initial Problem Statement is Too General
- 2 Synthesizing Nuances from the Literature
  - Nuance 1: Hydrophobic Interface Ordering
  - Nuance 2: Morphology and Confinement
  - Nuance 3: Tunable Interaction Spectrum
  - Nuance 4: Accurate Models
- 3 The Properly Defined Research Problem
- 4 Integrating Machine Learning and AI
  - Core ML Concept
  - ML Workflow
  - Advanced ML Implementations
- 5 Clear Goal and Total Refined Research Problem
- 6 Specific Aims: The Actionable Plan
- 7 Implementation Details: The Complete Pipeline
  - Phase 0: Configuration

# Outline II

- Phase 1: System Generation
- Phase 2: Simulation Execution
- Phase 3: Analysis

## 8 Technical Implementation Details

- Molecular Dynamics Engine
- Thermodynamic Ensemble Control
- Electrostatics: Ewald Summation
- Analysis Algorithms
- Machine Learning Algorithms

# The Starting Problem Statement

## Initial Question

“How does solvation structure change around a hydrophobic vs. hydrophilic nanoparticle?”

# The Starting Problem Statement

## Initial Question

“How does solvation structure change around a hydrophobic vs. hydrophilic nanoparticle?”

## Why This is Too General

This is an excellent entry point, but in cutting-edge research, it's considered general because it oversimplifies a complex phenomenon.

# Three Key Limitations

## ① Implies a Simple Binary

- Frames “hydrophobic” and “hydrophilic” as two distinct, opposite categories
- Literature shows it’s a complex spectrum
- The “hydrophilicity” of a metal surface  $\neq$  hydroxylated oxide surface

# Three Key Limitations

## ① Implies a Simple Binary

- Frames “hydrophobic” and “hydrophilic” as two distinct, opposite categories
- Literature shows it's a complex spectrum
- The “hydrophilicity” of a metal surface  $\neq$  hydroxylated oxide surface

## ② Neglects Morphology and Topology

- Assumes only variable is surface chemistry
- Shape, size, curvature (convex vs. concave), porosity dominate interaction
- Sometimes more important than intrinsic chemistry

# Three Key Limitations

## ① Implies a Simple Binary

- Frames “hydrophobic” and “hydrophilic” as two distinct, opposite categories
- Literature shows it’s a complex spectrum
- The “hydrophilicity” of a metal surface  $\neq$  hydroxylated oxide surface

## ② Neglects Morphology and Topology

- Assumes only variable is surface chemistry
- Shape, size, curvature (convex vs. concave), porosity dominate interaction
- Sometimes more important than intrinsic chemistry

## ③ Leaves “Solvation Structure” Undefined

- Could mean: density, orientation, H-bond lifetime, residence time, etc.
- Must define *which aspects* are investigated
- Must specify *which metrics* will quantify them

## Requirement

Build a more sophisticated problem statement that incorporates the nuances revealed by the literature.

We will extract specific, subtle findings from each paper to construct a more advanced research problem.

# Nuance 1: The Hydrophobic Interface is Not Empty

## Conventional View vs. Reality

- **Conventional:** Hydrophobic surface simply repels water
- **Reality:** Water reorganizes to preserve H-bond network
- Results in structures more ordered than bulk liquid

# Nuance 1: The Hydrophobic Interface is Not Empty

## Conventional View vs. Reality

- **Conventional:** Hydrophobic surface simply repels water
- **Reality:** Water reorganizes to preserve H-bond network
- Results in structures more ordered than bulk liquid

## Key Finding: Srivastava et al. [2024]

*“in smaller CNTs, water molecules adopt an icy structure near tube walls while maintaining liquid state towards the center.”*

# Nuance 1: The Hydrophobic Interface is Not Empty

## Conventional View vs. Reality

- **Conventional:** Hydrophobic surface simply repels water
- **Reality:** Water reorganizes to preserve H-bond network
- Results in structures more ordered than bulk liquid

## Key Finding: Srivastava et al. [2024]

*"in smaller CNTs, water molecules adopt an icy structure near tube walls while maintaining liquid state towards the center."*

## Paradigm Shift

- Reframes hydrophobic interface from zone of *depletion*
- To zone of *ice-like ordering*
- Key metric: **Tetrahedral order parameter**

## Nuance 2: Nanoparticle Morphology Creates Unique Environments

### Beyond Chemistry

- Chemical nature is not the only factor
- **Shape** dictates how solvent molecules can arrange
- Concave surfaces (pores, cages) behave fundamentally differently from convex ones

## Nuance 2: Nanoparticle Morphology Creates Unique Environments

### Beyond Chemistry

- Chemical nature is not the only factor
- **Shape** dictates how solvent molecules can arrange
- Concave surfaces (pores, cages) behave fundamentally differently from convex ones

### Key Finding: Gotzias [2022]

*“cyclohexane molecules remain attached on the concave surface of the nanotube or the nanocone without being disturbed by the water molecules entering the cavity.”*

# Nuance 2: Nanoparticle Morphology Creates Unique Environments

## Beyond Chemistry

- Chemical nature is not the only factor
- **Shape** dictates how solvent molecules can arrange
- Concave surfaces (pores, cages) behave fundamentally differently from convex ones

## Key Finding: Gotzias [2022]

*“cyclohexane molecules remain attached on the concave surface of the nanotube or the nanocone without being disturbed by the water molecules entering the cavity.”*

## Implication

- Interior of porous nanoparticle maintains hydrophobic environment
- Even when particle is immersed in water
- Driven by free energy

## Nuance 3: A Tunable Spectrum, Not Fixed Property

### Key Concept

- “Hydrophobic” and “hydrophilic” are endpoints of continuous spectrum
- This spectrum is called **wettability**
- Molecular-level interaction has direct consequences for macroscopic phenomena

## Nuance 3: A Tunable Spectrum, Not Fixed Property

### Key Concept

- “Hydrophobic” and “hydrophilic” are endpoints of continuous spectrum
- This spectrum is called **wettability**
- Molecular-level interaction has direct consequences for macroscopic phenomena

### Chen et al. [2014]

*“The interfacial thermal conductance is influenced by the selection of different water models and the interfacial wettability.”*

⇒ Links wettability to thermal properties

### Jorabchi et al. [2023]

*“the nanoalloys have less solvation energy in water than the other solvents. This is why the nanoalloys tend to approach more in this solvent”*

⇒ Links solvation energy to aggregation

## Nuance 4: Accurate Models Are Prerequisites

### Foundational Requirement

- Choice of potential function is not trivial
- It is **foundational** to simulation accuracy
- Applies to both solvent and solute

## Nuance 4: Accurate Models Are Prerequisites

### Foundational Requirement

- Choice of potential function is not trivial
- It is **foundational** to simulation accuracy
- Applies to both solvent and solute

### Water: Rick [2004]

*"The new model demonstrates a density maximum near 4°C, like the TIP5P model, and otherwise is similar to the TIP5P model for thermodynamic, dielectric, and dynamical properties of liquid water..."*

⇒ High-fidelity model needed

### Nanoparticle: Fronzi et al. [2023]

*"it is only for clusters with more than 30 atoms that interior gold atoms become present."*

⇒ Surface-to-volume ratio critical

# The Refined Research Problem (1/2)

## Overarching Goal

This research will conduct a systematic investigation into the **molecular-level determinants** of nanoparticle solvation in water, deconstructing “hydrophobicity” and “hydrophilicity” into a quantitative framework based on the **interplay of**:

- Surface chemistry
- Morphology
- Spatial confinement

# The Refined Research Problem (1/2)

## Overarching Goal

This research will conduct a systematic investigation into the **molecular-level determinants** of nanoparticle solvation in water, deconstructing “hydrophobicity” and “hydrophilicity” into a quantitative framework based on the **interplay of**:

- Surface chemistry
- Morphology
- Spatial confinement

## Grounded In

Grounded in the understanding that the hydrophobic interface can induce significant ordering [Srivastava et al., 2024], and that nanoparticle morphology creates distinct local solvation environments [Gotzias, 2022], this study will employ high-fidelity potentials for both water [Rick, 2004] and nanoparticles [Fronzi et al., 2023, Fomin, 2022] to address the following core questions.

# The Refined Research Problem (2/2)

## Core Questions

- ① How do quantitative structural metrics, specifically **tetrahedral order parameters**, differentiate “ice-like” ordering at non-polar carbon from layered, but more mobile, structure at metallic Ag/Au?
- ② To what extent does nanoparticle morphology control solvation? Specifically, how does water structure in **concave carbon nanotube** differ from **convex fullerene**, and how does this correlate with free energy of transfer?
- ③ How does continuous spectrum of **interfacial wettability** [Chen et al., 2014] translate to changes in first solvation shell dynamics (H-bond lifetimes, residence times)?
- ④ How do structural/dynamic signatures correlate with:
  - Thermodynamic drivers for aggregation [Jorabchi et al., 2023]
  - Interfacial heat transfer efficiency

# Elevation of the Project

## Key Achievement

This framing elevates the project from a **simple comparison** to a **deep, mechanistic study** of the fundamental physics governing the nanoparticle-water interface.

# Why Integrate ML/AI?

## The Traditional Approach

Simply correlating simulation results with macroscopic properties is valid but limited.

# Why Integrate ML/AI?

## The Traditional Approach

Simply correlating simulation results with macroscopic properties is valid but limited.

## The ML Advantage

Using ML opens a more novel and powerful way to:

- Analyze simulation data
- Leverage expensive computational results
- Make predictions without running new simulations

# The Core Idea: ML Learns the Physics

## The Computational Challenge

- MD simulations are computationally expensive
- Single 50 ns simulation for one  $\epsilon$  value can take days
- Want hydration number for new  $\epsilon$ ?  $\Rightarrow$  Another multi-day simulation

# The Core Idea: ML Learns the Physics

## The Computational Challenge

- MD simulations are computationally expensive
- Single 50 ns simulation for one  $\epsilon$  value can take days
- Want hydration number for new  $\epsilon$ ?  $\Rightarrow$  Another multi-day simulation

## The ML Solution

- Use expensive data points to **train a machine learning model**
- Model learns relationship between nanoparticle properties and solvation structure
- ML model becomes a **“surrogate model”**
- Highly efficient, data-driven approximation of expensive MD

# Step 1: Data Generation

## What You're Already Doing!

Perform MD simulations for carefully chosen input parameters.

# Step 1: Data Generation

## What You're Already Doing!

Perform MD simulations for carefully chosen input parameters.

### Inputs (Features):

- $\epsilon$  (LJ interaction strength)

### Outputs (Labels):

- Hydration Number
- RDF First Peak Height
- RDF First Peak Position

# Step 1: Data Generation

## What You're Already Doing!

Perform MD simulations for carefully chosen input parameters.

### Inputs (Features):

- $\epsilon$  (LJ interaction strength)

### Outputs (Labels):

- Hydration Number
- RDF First Peak Height
- RDF First Peak Position

## Example Values

Run simulations for  $\epsilon \in [0.02, 0.05, 0.1, 0.2, 0.5, 1.0]$  kcal/mol

Each with 3 replicas for statistics

# Example Dataset Structure

$\epsilon$ (kcal/mol)	Replica	Hydration Number	RDF Peak Height	RDF Peak Position ( $\text{\AA}$ )
0.02	1	3.1	1.8	3.4
0.02	2	3.3	1.9	3.4
0.02	3	3.2	1.8	3.5
0.05	1	4.5	2.5	3.6
...	...	...	...	...
1.00	3	12.5	5.1	3.9

Average replicas for each  $\epsilon$  to get final training dataset

## Step 2: Training the ML Surrogate Model

### Model Setup

Train a regression model to predict output properties from input features.

- **Model Input (X):**  $\epsilon$  values
- **Model Output (Y):** Hydration number, RDF peak height, etc.

## Step 2: Training the ML Surrogate Model

### Model Setup

Train a regression model to predict output properties from input features.

- **Model Input (X):**  $\epsilon$  values
- **Model Output (Y):** Hydration number, RDF peak height, etc.

### Model Options (using Scikit-learn)

- 1 **Random Forest Regressor:** Robust for small datasets, less prone to overfitting
- 2 **Gradient Boosting (XGBoost/LightGBM):** State-of-the-art on tabular data
- 3 **Gaussian Process Regressor:** Provides prediction + uncertainty estimate
- 4 **Simple Neural Network:** For “AI” flavor (Keras/TensorFlow, 2 hidden layers, 32 neurons each)

## Step 2: Training the ML Surrogate Model

### Model Setup

Train a regression model to predict output properties from input features.

- **Model Input (X):**  $\epsilon$  values
- **Model Output (Y):** Hydration number, RDF peak height, etc.

### Model Options (using Scikit-learn)

- 1 **Random Forest Regressor:** Robust for small datasets, less prone to overfitting
- 2 **Gradient Boosting (XGBoost/LightGBM):** State-of-the-art on tabular data
- 3 **Gaussian Process Regressor:** Provides prediction + uncertainty estimate
- 4 **Simple Neural Network:** For “AI” flavor (Keras/TensorFlow, 2 hidden layers, 32 neurons each)

Split data: Train on subset, test on held-out values

## Step 3: Prediction and Validation

### The “Aha!” Moment

- 1 **Predict:** Choose  $\varepsilon$  not in training data (e.g.,  $\varepsilon = 0.1$ )
- 2 **ML Prediction:** Model gives instant answer:  
*“For  $\varepsilon = 0.1$ , hydration number =  $6.8 \pm 0.2$ ”*
- 3 **Validate:** Run actual MD simulation for  $\varepsilon = 0.1$  (takes days)  
True answer: **6.9**
- 4 **Success!** ML predicted complex physical simulation in **seconds**, saving days of compute time

# Reframing the Research Problem

## Updated Problem Statement

“How does solvation structure change around a hydrophobic vs. hydrophilic nanoparticle? This research will address this by:

- 1 Generating high-fidelity molecular dynamics data for a range of nanoparticle interaction strengths ( $\epsilon$ )
- 2 Using this data to train a machine learning surrogate model capable of instantly and accurately predicting key structural metrics

**Ultimate goal:** Create a predictive framework that replaces expensive first-principles simulation with rapid, data-driven model, enabling efficient exploration of the hydrophobic-to-hydrophilic transition.”

## Beyond Scalar Properties

**Challenge:** Instead of predicting single number (peak height), predict entire function

**How:** Train neural network where:

- Input:  $\varepsilon$  (single value)
- Output: Vector of 200 numbers representing  $g(r)$  at each point  $r$

# Advanced ML: Predicting Entire RDF Curve

## Beyond Scalar Properties

**Challenge:** Instead of predicting single number (peak height), predict entire function

**How:** Train neural network where:

- Input:  $\varepsilon$  (single value)
- Output: Vector of 200 numbers representing  $g(r)$  at each point  $r$

## Impact

- More complex but far more powerful
- Captures complete structural information
- Enables detailed analysis without running MD

# Advanced ML: Unsupervised Discovery of Water States

## Approach

- ① Extract thousands of snapshots of water molecules in first solvation shell
- ② For each water molecule, create feature vector:
  - Distance from surface
  - Tetrahedral order parameter  $S_q$
  - Orientation of dipole
- ③ Apply clustering algorithm (DBSCAN or k-Means) on this dataset

# Advanced ML: Unsupervised Discovery of Water States

## Approach

- ① Extract thousands of snapshots of water molecules in first solvation shell
- ② For each water molecule, create feature vector:
  - Distance from surface
  - Tetrahedral order parameter  $S_q$
  - Orientation of dipole
- ③ Apply clustering algorithm (DBSCAN or k-Means) on this dataset

## Discovery

Algorithm automatically discovers distinct “states” of interfacial water:

- “Ice-like”
- “Bulk-like”
- “Disordered”

Without being told what to look for!

# Advanced ML: Generative AI for Solvation Shells

## State-of-the-Art Approach

**How:** Train generative model (VAE or diffusion model) on simulation snapshots

**Goal:** Given  $\epsilon$  value, model *generates* 3D configuration of most probable water structure around nanoparticle

# Advanced ML: Generative AI for Solvation Shells

## State-of-the-Art Approach

**How:** Train generative model (VAE or diffusion model) on simulation snapshots

**Goal:** Given  $\epsilon$  value, model *generates* 3D configuration of most probable water structure around nanoparticle

## Revolutionary Impact

- Creates “snapshot” without running simulation at all
- Transforms project from *descriptive* to *generative*
- Opens door to rapid exploration of parameter space

# Advanced ML: Generative AI for Solvation Shells

## State-of-the-Art Approach

**How:** Train generative model (VAE or diffusion model) on simulation snapshots

**Goal:** Given  $\epsilon$  value, model *generates* 3D configuration of most probable water structure around nanoparticle

## Revolutionary Impact

- Creates “snapshot” without running simulation at all
- Transforms project from *descriptive* to *generative*
- Opens door to rapid exploration of parameter space

## Paradigm Shift

By integrating ML, you transform project from one that **describes** a phenomenon to one that **predicts** it.

# The Clear Goal of the Whole Study

## Overarching Goal

Develop a **predictive, data-driven framework** for understanding and modeling the molecular structure of water at nanoparticle interfaces.

# The Clear Goal of the Whole Study

## Overarching Goal

Develop a **predictive, data-driven framework** for understanding and modeling the molecular structure of water at nanoparticle interfaces.

## Beyond Traditional Simulations

- Move beyond descriptive one-off simulations
- Use high-fidelity MD as “ground truth”
- Train suite of machine learning models
- Deliverable: Computationally inexpensive **“MD-ML surrogate”**
- Can instantly predict complex, multi-scale solvation structure
- Autonomously discover fundamental physical states of interfacial water

# Total Refined Research Problem (1/3)

## Fundamental Question

How does water structure itself at the interface of hydrophobic and hydrophilic nanoparticles?

# Total Refined Research Problem (1/3)

## Fundamental Question

How does water structure itself at the interface of hydrophobic and hydrophilic nanoparticles?

## Central Hypothesis

The traditional hydrophobic/hydrophilic dichotomy is an **insufficient descriptor**. True solvation structure arises from complex interplay of:

- Surface chemistry
- Nanoparticle morphology
- Spatial confinement

# Total Refined Research Problem (2/3)

## Methodology: Hybrid MD-ML Approach

### Foundational Dataset:

- High-fidelity MD simulations
- Spherical LJ solutes with systematically varied  $\epsilon$
- Using structurally accurate TIP5P-EW water model

### Data Usage:

- Not merely for descriptive analysis
- Train and validate hierarchy of ML models
- Models designed to *learn the underlying physics of solvation*

# Total Refined Research Problem (3/3)

## Specific Components

- ① Develop **deep learning regressor** to predict entire RDF as continuous function of  $\varepsilon$
- ② Employ **unsupervised clustering** on molecular-level features (including tetrahedral order) to autonomously identify distinct states of interfacial water
- ③ (Future) Explore **generative AI models** to construct realistic 3D solvation shell configurations from input parameters

# Total Refined Research Problem (3/3)

## Specific Components

- 1 Develop **deep learning regressor** to predict entire RDF as continuous function of  $\varepsilon$
- 2 Employ **unsupervised clustering** on molecular-level features (including tetrahedral order) to autonomously identify distinct states of interfacial water
- 3 (Future) Explore **generative AI models** to construct realistic 3D solvation shell configurations from input parameters

## Scientific Questions to Answer

- Nature of interfacial ordering
- Dominant role of nanoparticle morphology
- Dynamics of first solvation shell
- Thermodynamic drivers of aggregation and heat transfer

# Overview of Three Aims

- ① **Aim 1:** Establish “Ground Truth” Dataset via High-Fidelity MD
- ② **Aim 2:** Develop Predictive ML Surrogate Model for Rapid Structural Prediction
- ③ **Aim 3:** Discover Latent Solvation States using Unsupervised Learning

# Aim 1: Ground Truth Dataset I

## Action

Perform core MD simulations:

- Series of runs with spherical LJ solutes in TIP5P-EW water
- Sweep  $\varepsilon$  parameter from highly hydrophobic to highly hydrophilic

# Aim 1: Ground Truth Dataset II

## Output

For each simulation:

- Full RDF curves
- Hydration numbers
- Trajectories with detailed molecular information:
  - Positions
  - Orientations
  - Tetrahedral order parameters for every water molecule near interface

## Purpose

This dataset is the foundational input for Aims 2 and 3

## Aim 2: Predictive ML Surrogate Model

### Action

Use dataset from Aim 1 to train neural network:

- **Input:** Nanoparticle's  $\varepsilon$  value
- **Output:** Predicted 200-point vector representing entire  $g(r)$  curve

## Aim 2: Predictive ML Surrogate Model

### Action

Use dataset from Aim 1 to train neural network:

- **Input:** Nanoparticle's  $\varepsilon$  value
- **Output:** Predicted 200-point vector representing entire  $g(r)$  curve

### Validation

Model validated by ability to accurately predict RDF for  $\varepsilon$  values held out from training set

## Aim 2: Predictive ML Surrogate Model

### Action

Use dataset from Aim 1 to train neural network:

- **Input:** Nanoparticle's  $\varepsilon$  value
- **Output:** Predicted 200-point vector representing entire  $g(r)$  curve

### Validation

Model validated by ability to accurately predict RDF for  $\varepsilon$  values held out from training set

### Goal

Create a tool that can generate physically accurate RDF in **seconds**, bypassing need for multi-day MD simulation

## Aim 3: Discover Latent Solvation States

### Action

- ➊ From trajectories (Aim 1), extract thousands of snapshots of individual water molecules in first solvation shell
- ➋ For each molecule, create feature vector:
  - Distance from surface
  - $S_q$  value (tetrahedral order)
  - Number of hydrogen bonds
- ➌ Apply clustering algorithm (k-Means or DBSCAN) to high-dimensional dataset

## Aim 3: Discover Latent Solvation States

### Action

- 1 From trajectories (Aim 1), extract thousands of snapshots of individual water molecules in first solvation shell
- 2 For each molecule, create feature vector:
  - Distance from surface
  - $S_q$  value (tetrahedral order)
  - Number of hydrogen bonds
- 3 Apply clustering algorithm (k-Means or DBSCAN) to high-dimensional dataset

### Goal

Allow machine to autonomously discover fundamental “states” of interfacial water:

- Provides data-driven answer to what “ice-like” vs. “disordered” vs. “bulk-like” truly means
- Quantify population of these states as function of  $\varepsilon$

# The Powerful Narrative

## Why This Structure Works

- **MD simulations** provide essential physical accuracy
- **ML models** provide novel predictive power
- **Deep analytical insight** not possible with simulation-only study
- Enables tackling core scientific questions in fundamentally new way

# The Powerful Narrative

## Why This Structure Works

- **MD simulations** provide essential physical accuracy
- **ML models** provide novel predictive power
- **Deep analytical insight** not possible with simulation-only study
- Enables tackling core scientific questions in fundamentally new way

## Impact

Transforms computational chemistry from expensive black-box calculations to intelligent, data-driven prediction

# High-Level Pipeline Overview

- ① **Phase 0: Configuration & Setup**
  - Define all parameters
  - Set up environment
- ② **Phase 1: System Generation (One-time)**
  - Create initial configuration
  - Convert to LAMMPS format
- ③ **Phase 2: Simulation Workflow (Automated)**
  - Generate input files for all  $\epsilon$  values
  - Run simulations
- ④ **Phase 3: Analysis**
  - Process RDF data
  - Calculate coordination numbers
  - Train ML models

# Phase 0.1: Environment Setup

## Required Software

- 1 **LAMMPS:** Compiled with MOLECULE and KSPACE packages for TIP5P support
- 2 **PACKMOL:** For generating initial configurations
- 3 **VMD:** For file format conversions
- 4 **Python environment:** With scientific computing libraries

# Phase 0.1: Environment Setup

## Required Software

- ① **LAMMPS:** Compiled with MOLECULE and KSPACE packages for TIP5P support
- ② **PACKMOL:** For generating initial configurations
- ③ **VMD:** For file format conversions
- ④ **Python environment:** With scientific computing libraries

## Python Setup

```
bash scripts/setup_python_env.sh  
source .venv/bin/activate
```

## Phase 0.2: Master Configuration File

### configs/params.yaml

Single source of truth for all parameters:

- Water model: TIP5P-EW (5-site, Ewald-optimized)
- Solute: LJ sphere with varied  $\epsilon$
- $\epsilon$  sweep: [0.02, 0.05, 0.1, 0.2, 0.5, 1.0] kcal/mol
- Box size: 60 Å cubic
- Number of replicas: 3 per  $\epsilon$
- Equilibration: 5 ns
- Production: 50 ns
- Timestep: 2.0 fs

## Phase 0.3: LAMMPS Input Template

### `in/cg_sphere.in.template`

Template handles TIP5P-EW correctly:

- `pair_style lj/cut/tip4p/long`
- Proper treatment of rigid water molecules
- Immobilized solute at origin
- Three stages:
  - 1 Energy minimization
  - 2 NPT equilibration (300K, 1 atm)
  - 3 NVT production run
- RDF computation during production
- Trajectory output for analysis

# Phase 1: One-Time System Building I

## Step 1.1: Generate PACKMOL Input

```
python3 tools/packmol_wrapper.py
```

Calculates  $\sim 7200$  water molecules needed for  $60\text{\AA}$  box

# Phase 1: One-Time System Building II

## Step 1.4: Convert to LAMMPS Format

```
vmd -dispdev text -e tools/solvate_vmd.tcl
```

Creates critical data/system.data file

## Phase 2: Automated Simulation Workflow

### Step 2.1: Generate All Input Files

```
python3 scripts/sweep_eps.py
```

Creates 18 LAMMPS input files ( $6 \epsilon \times 3$  replicas)

Directory structure: experiments/eps\_X/replica\_Y/run.in

## Phase 2: Automated Simulation Workflow

### Step 2.1: Generate All Input Files

```
python3 scripts/sweep_eps.py
```

Creates 18 LAMMPS input files ( $6 \epsilon \times 3$  replicas)

Directory structure: experiments/eps\_X/replica\_Y/run.in

### Step 2.2: Run Simulations

**CPU:** mpirun -np 1 lmp\_mpi -in run.in

**GPU:** lmp -k on g 1 -in run.in

**Expected runtime:** ~4-8 hours on modern GPU (A40/A6000)

## Phase 2: Automated Simulation Workflow

### Step 2.1: Generate All Input Files

```
python3 scripts/sweep_eps.py
```

Creates 18 LAMMPS input files ( $6 \epsilon \times 3$  replicas)

Directory structure: experiments/eps\_X/replica\_Y/run.in

### Step 2.2: Run Simulations

**CPU:** mpirun -np 1 lmp\_mpi -in run.in

**GPU:** lmp -k on g 1 -in run.in

**Expected runtime:**  $\sim 4\text{-}8$  hours on modern GPU (A40/A6000)

### Total Simulation Time

18 simulations  $\times$  6 hours =  $\sim 108$  hours (4.5 days) if run sequentially

Much faster if parallelized across multiple GPUs

## Phase 3.1: RDF Computation

### During Simulation

LAMMPS computes RDF during production run:

- `compute rdf_run SOLUTE OXYGEN rdf`
- `fix ave/time` accumulates time averages
- Output: `rdf_solute_0.dat`

## Phase 3.2: Coordination Number Calculation

### Integration of RDF

Calculate hydration number by integrating RDF to first minimum:

$$N = 4\pi\rho \int_0^{r_{\min}} g(r)r^2 dr$$

where:

- $N$  = coordination number
- $\rho$  = bulk water density
- $r_{\min}$  = position of first minimum in RDF

## Phase 3.2: Coordination Number Calculation

### Integration of RDF

Calculate hydration number by integrating RDF to first minimum:

$$N = 4\pi\rho \int_0^{r_{\min}} g(r)r^2 dr$$

where:

- $N$  = coordination number
- $\rho$  = bulk water density
- $r_{\min}$  = position of first minimum in RDF

### Python Implementation

```
python3 analysis/compute_rdf.py
```

Parses LAMMPS output and calculates  $N$  for each replica

## Phase 3.3: Statistical Analysis and Visualization

### Aggregate Results

- 1 Average hydration number across 3 replicas for each  $\varepsilon$
- 2 Calculate standard deviation for error bars
- 3 Create publication-quality plot

## Phase 3.3: Statistical Analysis and Visualization

### Aggregate Results

- 1 Average hydration number across 3 replicas for each  $\varepsilon$
- 2 Calculate standard deviation for error bars
- 3 Create publication-quality plot

### Final Output

Plot of hydration number vs.  $\varepsilon$  showing:

- Transition from hydrophobic to hydrophilic regime
- Statistical uncertainty from replicas
- Clear trend in solvation structure

This plot is the primary scientific result of foundational MD phase

## Phase 3.4: ML Model Training

### Using the Generated Data

- ① Load aggregated dataset (averaged over replicas)
- ② Split into training and test sets
- ③ Train chosen ML model (Random Forest, XGBoost, or Neural Network)
- ④ Validate on held-out  $\varepsilon$  values
- ⑤ Generate predictions for interpolated/extrapolated values

## Phase 3.4: ML Model Training

### Using the Generated Data

- 1 Load aggregated dataset (averaged over replicas)
- 2 Split into training and test sets
- 3 Train chosen ML model (Random Forest, XGBoost, or Neural Network)
- 4 Validate on held-out  $\varepsilon$  values
- 5 Generate predictions for interpolated/extrapolated values

### Success Metric

ML model should predict hydration number (or full RDF) for unseen  $\varepsilon$  with error comparable to statistical uncertainty from MD replicas

# Overview of Technical Implementation

## Five Key Algorithmic Components

- ① **MD Engine:** Newton's equations integration
- ② **Thermodynamic Control:** Temperature and pressure regulation
- ③ **Electrostatics:** Ewald summation for long-range forces
- ④ **Analysis:** RDF computation and coordination numbers
- ⑤ **Machine Learning:** Neural networks and clustering

# Conclusion: Key Takeaways

## Research Problem Transformation

- **From:** Simple binary classification (hydrophobic vs. hydrophilic)
- **To:** Complex interplay of surface chemistry, morphology, and spatial confinement
- **Grounded in:** Literature synthesis from 7 key papers

# Conclusion: Key Takeaways

## Research Problem Transformation

- **From:** Simple binary classification (hydrophobic vs. hydrophilic)
- **To:** Complex interplay of surface chemistry, morphology, and spatial confinement
- **Grounded in:** Literature synthesis from 7 key papers

## Methodological Innovation

- **Hybrid MD-ML Framework:** High-fidelity simulations + machine learning
- **Predictive Capability:** Instant RDF prediction from  $\varepsilon$  values
- **Autonomous Discovery:** Unsupervised identification of interfacial water states

# Conclusion: Key Takeaways

## Research Problem Transformation

- **From:** Simple binary classification (hydrophobic vs. hydrophilic)
- **To:** Complex interplay of surface chemistry, morphology, and spatial confinement
- **Grounded in:** Literature synthesis from 7 key papers

## Methodological Innovation

- **Hybrid MD-ML Framework:** High-fidelity simulations + machine learning
- **Predictive Capability:** Instant RDF prediction from  $\varepsilon$  values
- **Autonomous Discovery:** Unsupervised identification of interfacial water states

## Scientific Impact

- **Theoretical:** Mechanistic understanding of nanoparticle-water interfaces
- **Practical:** Orders-of-magnitude speedup in parameter space exploration

# Conclusion: Key Takeaways

## Research Problem Transformation

- **From:** Simple binary classification (hydrophobic vs. hydrophilic)
- **To:** Complex interplay of surface chemistry, morphology, and spatial confinement
- **Grounded in:** Literature synthesis from 7 key papers

## Methodological Innovation

- **Hybrid MD-ML Framework:** High-fidelity simulations + machine learning
- **Predictive Capability:** Instant RDF prediction from  $\varepsilon$  values
- **Autonomous Discovery:** Unsupervised identification of interfacial water states

## Scientific Impact

- **Theoretical:** Mechanistic understanding of nanoparticle-water interfaces
- **Practical:** Orders-of-magnitude speedup in parameter space exploration

# References I

- Xiaoling Chen, Antonio Munjiza, Kai Zhang, and Dongsheng Wen. Molecular dynamics simulation of heat transfer from a gold nanoparticle to a water pool. *The Journal of Physical Chemistry C*, 118(2):1285–1293, 2014. doi: 10.1021/jp410054j.
- Yu. D. Fomin. Molecular simulation of the formation of carbon nanoparticles. *Nanobiotechnology Reports*, 17(4):462–466, 2022. doi: 10.1134/S2635167622040097.
- Marco Fronzi, Roger D. Amos, and Rika Kobayashi. Evaluation of machine learning interatomic potentials for gold nanoparticles—transferability towards bulk. *Nanomaterials*, 13(12):1832, 2023. doi: 10.3390/nano13121832.
- Anastasios Gotzias. Umbrella sampling simulations of carbon nanoparticles crossing immiscible solvents. *Molecules*, 27(3):956, 2022. doi: 10.3390/molecules27030956.
- Majid Namayandeh Jorabchi, Mohsen Abbaspour, Elaheh K. Goharshadi, Iman Salahshoori, and Sebastian Wohlrab. Molecular dynamics simulation of Pt@Au nanoalloy in various solvents: Investigation of solvation, aggregation, and possible coalescence. *Journal of Materials Research and Technology*, 26:2863–2880, 2023. doi: 10.1016/j.jmrt.2023.08.091.

## References II

- Steven W. Rick. A reoptimization of the five-site water potential (TIP5P) for use with Ewald sums. *The Journal of Chemical Physics*, 120(13):6085–6093, 2004. doi: 10.1063/1.1652434.
- Amit Srivastava, Sufian Abedrabbo, Jamal Hassan, and Dirar Homouz. Dynamics of confined water inside carbon nanotubes based on studying tetrahedral order parameters. *Scientific Reports*, 14(1):15480, 2024. doi: 10.1038/s41598-024-66317-1.

# The Potential Energy Function

## Total Energy of the System

The potential energy function describes the total energy given positions of all  $N$  atoms:

$$U(\mathbf{r}^N) = \sum_{i < j} U_{LJ}(r_{ij}) + \sum_{i < j} U_{Coulomb}(r_{ij})$$

# The Potential Energy Function

## Total Energy of the System

The potential energy function describes the total energy given positions of all  $N$  atoms:

$$U(\mathbf{r}^N) = \sum_{i < j} U_{LJ}(r_{ij}) + \sum_{i < j} U_{Coulomb}(r_{ij})$$

## Lennard-Jones Potential

Describes short-range repulsion and van der Waals attraction:

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

# The Potential Energy Function

## Total Energy of the System

The potential energy function describes the total energy given positions of all  $N$  atoms:

$$U(\mathbf{r}^N) = \sum_{i < j} U_{LJ}(r_{ij}) + \sum_{i < j} U_{Coulomb}(r_{ij})$$

## Lennard-Jones Potential

Describes short-range repulsion and van der Waals attraction:

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

## Coulomb Potential

Describes electrostatic interactions between point charges:

# From Energy to Forces

## Force Calculation

The force on atom  $i$  is the negative gradient of potential energy:

$$\mathbf{F}_i = -\nabla_{\mathbf{r}_i} U(\mathbf{r}^N)$$

# From Energy to Forces

## Force Calculation

The force on atom  $i$  is the negative gradient of potential energy:

$$\mathbf{F}_i = -\nabla_{\mathbf{r}_i} U(\mathbf{r}^N)$$

## Computational Bottleneck

This requires summing forces from all other relevant particles - the most expensive part of MD simulation.

# Velocity Verlet Integration

## Time Integration Algorithm

Given positions  $\mathbf{r}(t)$ , velocities  $\mathbf{v}(t)$ , accelerations  $\mathbf{a}(t)$  at time  $t$ :

**Step 1:** Calculate new positions and half-step velocities

$$\mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \mathbf{v}(t)\Delta t + \frac{1}{2}\mathbf{a}(t)\Delta t^2$$

$$\mathbf{v}(t + \frac{1}{2}\Delta t) = \mathbf{v}(t) + \frac{1}{2}\mathbf{a}(t)\Delta t$$

# Velocity Verlet Integration

## Time Integration Algorithm

Given positions  $\mathbf{r}(t)$ , velocities  $\mathbf{v}(t)$ , accelerations  $\mathbf{a}(t)$  at time  $t$ :

**Step 1:** Calculate new positions and half-step velocities

$$\mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \mathbf{v}(t)\Delta t + \frac{1}{2}\mathbf{a}(t)\Delta t^2$$

$$\mathbf{v}(t + \frac{1}{2}\Delta t) = \mathbf{v}(t) + \frac{1}{2}\mathbf{a}(t)\Delta t$$

## Step 2:

Calculate forces and new accelerations

$$\mathbf{a}(t + \Delta t) = \frac{\mathbf{F}(t + \Delta t)}{m}$$

# Velocity Verlet Integration

## Time Integration Algorithm

Given positions  $\mathbf{r}(t)$ , velocities  $\mathbf{v}(t)$ , accelerations  $\mathbf{a}(t)$  at time  $t$ :

**Step 1:** Calculate new positions and half-step velocities

$$\mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \mathbf{v}(t)\Delta t + \frac{1}{2}\mathbf{a}(t)\Delta t^2$$

$$\mathbf{v}(t + \frac{1}{2}\Delta t) = \mathbf{v}(t) + \frac{1}{2}\mathbf{a}(t)\Delta t$$

## Step 2:

Calculate forces and new accelerations

$$\mathbf{a}(t + \Delta t) = \frac{\mathbf{F}(t + \Delta t)}{m}$$

## Step 3:

# Nosé-Hoover Thermostat (NVT)

## Temperature Control

Introduces additional degree of freedom  $\xi$  (thermal reservoir):

$$\begin{aligned}\dot{\mathbf{p}}_i &= \mathbf{F}_i - \xi \mathbf{p}_i \\ \dot{\xi} &= \frac{1}{Q} \left( \sum_i \frac{\mathbf{p}_i^2}{m_i} - g k_B T_{target} \right)\end{aligned}$$

# Nosé-Hoover Thermostat (NVT)

## Temperature Control

Introduces additional degree of freedom  $\xi$  (thermal reservoir):

$$\dot{\mathbf{p}}_i = \mathbf{F}_i - \xi \mathbf{p}_i$$
$$\dot{\xi} = \frac{1}{Q} \left( \sum_i \frac{\mathbf{p}_i^2}{m_i} - g k_B T_{target} \right)$$

- $Q$ : Thermostat "mass" (controls fluctuation frequency)
- $g$ : Number of degrees of freedom
- If system too hot:  $\xi$  increases, friction term cools system

# Rigid Body Algorithm for Water

## Fix rigid/small in LAMMPS

For rigid water models (TIP5P-EW), treats each molecule as single entity:

- 1 **Compute Forces:** Total force  $\mathbf{F}_{total}$  and torque  $\tau_{total}$  on center of mass
- 2 **Integrate Motion:** Update translational/rotational velocity of molecule as whole
- 3 **Update Atoms:** Calculate individual atom positions from new center of mass position/orientation

# Rigid Body Algorithm for Water

## Fix rigid/small in LAMMPS

For rigid water models (TIP5P-EW), treats each molecule as single entity:

- 1 **Compute Forces:** Total force  $\mathbf{F}_{total}$  and torque  $\tau_{total}$  on center of mass
- 2 **Integrate Motion:** Update translational/rotational velocity of molecule as whole
- 3 **Update Atoms:** Calculate individual atom positions from new center of mass position/orientation

## Why Rigid?

2.0 fs timestep would break water bonds without rigid constraints - this is more efficient than SHAKE.

# The Ewald Summation Problem

## Challenge

Coulomb potential  $1/r$  decays slowly. In periodic systems, atoms interact with infinite periodic images.

# The Ewald Summation Problem

## Challenge

Coulomb potential  $1/r$  decays slowly. In periodic systems, atoms interact with infinite periodic images.

## Solution: Ewald Summation

Split interaction into short-range and long-range parts using Gaussian screening.

# Ewald Splitting

## Mathematical Splitting

$$\frac{1}{r} = \underbrace{\frac{\operatorname{erfc}(\alpha r)}{r}}_{\text{Short-Range}} + \underbrace{\frac{\operatorname{erf}(\alpha r)}{r}}_{\text{Long-Range}}$$

# Ewald Splitting

## Mathematical Splitting

$$\frac{1}{r} = \underbrace{\frac{\text{erfc}(\alpha r)}{r}}_{\text{Short-Range}} + \underbrace{\frac{\text{erf}(\alpha r)}{r}}_{\text{Long-Range}}$$

- **Short-range:** Decays rapidly, computed in real space with cutoff
- **Long-range:** Smooth Gaussians, computed efficiently in reciprocal space via FFT

# Ewald Splitting

## Mathematical Splitting

$$\frac{1}{r} = \underbrace{\frac{\text{erfc}(\alpha r)}{r}}_{\text{Short-Range}} + \underbrace{\frac{\text{erf}(\alpha r)}{r}}_{\text{Long-Range}}$$

- **Short-range:** Decays rapidly, computed in real space with cutoff
- **Long-range:** Smooth Gaussians, computed efficiently in reciprocal space via FFT

## PME Algorithm (pppm/tip4p)

- ① Charge assignment to 3D grid
- ② FFT to solve Poisson equation
- ③ Inverse FFT for potential
- ④ Force interpolation back to atoms

# Radial Distribution Function (RDF)

## Computational Implementation

RDF calculated via histogram method:

- 1 Create histogram array of  $N_{bins}$  counters
- 2 For each solute-water pair: calculate  $r_{ij}$ , find bin  $\lfloor r_{ij}/\Delta r \rfloor$
- 3 Increment histogram[bin]++
- 4 Normalize with spherical shell volume factor:

$$g(r_i) = \frac{\text{histogram}[i]}{\text{num\_timesteps} \times \rho_{bulk} \times 4\pi r_i^2 \Delta r}$$

# Coordination Number Calculation

## Numerical Integration

Hydration number via trapezoidal rule integration:

$$N_{coord} = 4\pi\rho \int_0^{r_{min}} g(r)r^2 dr$$

# Coordination Number Calculation

## Numerical Integration

Hydration number via trapezoidal rule integration:

$$N_{coord} = 4\pi\rho \int_0^{r_{min}} g(r)r^2 dr$$

Trapezoidal rule approximation:

$$\int_0^{r_{min}} f(r)dr \approx \sum_{k=1}^M \frac{f(r_k) + f(r_{k-1})}{2} (r_k - r_{k-1})$$

Where  $f(r) = g(r)r^2$

# ML Goal 1: Predicting RDF with Neural Network

## Problem Formulation

Supervised regression:  $f : \mathbb{R} \rightarrow \mathbb{R}^{200}$

- **Input:**  $\epsilon$  (scalar)
- **Output:**  $\mathbf{g} = [g(r_1), g(r_2), \dots, g(r_{200})]$  (200-dim vector)

# ML Goal 1: Predicting RDF with Neural Network

## Problem Formulation

Supervised regression:  $f : \mathbb{R} \rightarrow \mathbb{R}^{200}$

- **Input:**  $\epsilon$  (scalar)
- **Output:**  $\mathbf{g} = [g(r_1), g(r_2), \dots, g(r_{200})]$  (200-dim vector)

## Multi-Layer Perceptron Architecture

- Input layer: 1 neuron ( $\epsilon$ )
- Hidden layers: 3 layers  $\times$  64 neurons each
- Output layer: 200 neurons (RDF vector)
- Activation: ReLU function  $\phi(x) = \max(0, x)$

# Neural Network Training

## Forward Propagation

Neuron output:  $a_j = \phi(\sum_i w_{ij}a_i + b_j)$

# Neural Network Training

## Forward Propagation

Neuron output:  $a_j = \phi(\sum_i w_{ij} a_i + b_j)$

## Loss Function

Mean Squared Error between prediction  $\hat{\mathbf{g}}$  and true RDF  $\mathbf{g}$ :

$$L = \frac{1}{200} \sum_{i=1}^{200} (g_i - \hat{g}_i)^2$$

# Neural Network Training

## Forward Propagation

Neuron output:  $a_j = \phi(\sum_i w_{ij}a_i + b_j)$

## Loss Function

Mean Squared Error between prediction  $\hat{\mathbf{g}}$  and true RDF  $\mathbf{g}$ :

$$L = \frac{1}{200} \sum_{i=1}^{200} (g_i - \hat{g}_i)^2$$

## Optimization

Adam optimizer minimizes loss via backpropagation:

$\nabla L \rightarrow$  update weights in direction of steepest descent

# ML Goal 2: Discovering Water States

## Unsupervised k-Means Clustering

Group  $M$  water molecules into  $k$  clusters based on feature vectors.

### Algorithm:

- 1 Initialize  $k$  random centroids  $\{\mathbf{c}_1, \dots, \mathbf{c}_k\}$
- 2 **Assignment:**  $\text{cluster}(\mathbf{x}_i) = \arg \min_j \|\mathbf{x}_i - \mathbf{c}_j\|^2$
- 3 **Update:**  $\mathbf{c}_j = \frac{1}{|S_j|} \sum_{\mathbf{x}_i \in S_j} \mathbf{x}_i$
- 4 Repeat until convergence

# ML Goal 2: Discovering Water States

## Unsupervised k-Means Clustering

Group  $M$  water molecules into  $k$  clusters based on feature vectors.

### Algorithm:

- 1 Initialize  $k$  random centroids  $\{\mathbf{c}_1, \dots, \mathbf{c}_k\}$
- 2 **Assignment:**  $\text{cluster}(\mathbf{x}_i) = \arg \min_j \|\mathbf{x}_i - \mathbf{c}_j\|^2$
- 3 **Update:**  $\mathbf{c}_j = \frac{1}{|S_j|} \sum_{\mathbf{x}_i \in S_j} \mathbf{x}_i$
- 4 Repeat until convergence

## Features

Distance, tetrahedral order  $S_q$ , orientation, H-bond count

