Development of large-scale molecular and nanomaterial models

Mikhail Ivanov
Development of large-scale molecular and nanomaterial models

Mikhail Ivanov

Academic dissertation for the Degree of Doctor of Philosophy in Physical Chemistry at Stockholm University to be publicly defended on Friday 3 May 2024 at 14.00 in Magnélsalen, Kemiska övningslaboratoriet, Svante Arrhenius väg 16 B and online via Zoom, public link is available at the department website.

Abstract
Molecular simulations can access unique atomic-scale information about new materials, pharmaceuticals, and biological environments, making cost-effective predictions and aiding experimental studies. They are particularly useful for describing the mechanisms of nanoscale phenomena and the biological/inorganic interfaces. However, the computational cost of molecular simulations increases with the size of the system as well as with the model complexity, which is related to the accuracy of the simulation. This thesis aims to develop efficient large-scale molecular models that capture important structural details of the atomistic simulations. In particular, we focus on the TiO₂-lipid interface, which forms in the living cells, exposed to TiO₂ nanomaterials, but is also relevant in the context of biomedical applications. We have studied the interface using atomistic molecular dynamics simulations and found that the characteristics of the lipid adsorption depend on the type of the TiO₂ surface, lipid headgroup composition, and the presence of cholesterol. We then derive a coarse-grained molecular model of the TiO₂-lipid interface to enable the large-scale simulations of TiO₂ nanoparticles interacting with model cell membranes. We show that the strength of the lipid adsorption increases with the size of the nanoparticle and that a small TiO₂ nanoparticle can become partially wrapped by a lipid membrane. To improve the transferability of the coarse-grained model, we design and test an artificial neural network that learns the interactions in coarse-grained water-methanol solutions from the structural data obtained in multiple reference simulations at atomistic resolution. We show that in the studied system, the neural network learns the many-body interactions and accurately reproduces the structural properties of the solution at different concentrations.

Keywords: Molecular simulations, Coarse-grained models, Lipids, TiO2 surface, Machine learning.

Stockholm 2024
http://urn.kb.se/resolve?urn=urn:nbn:se:su:diva-227287
ISBN 978-91-8014-705-7
ISBN 978-91-8014-706-4

Department of Materials and Environmental Chemistry (MMK)
Stockholm University, 106 91 Stockholm
DEVELOPMENT OF LARGE-SCALE MOLECULAR AND NANOMATERIAL MODELS

Mikhail Ivanov
Development of large-scale molecular and nanomaterial models

Mikhail Ivanov
To those by my side
Pelorat said, "How long does all that take?"
"Not very long. The computer does the heavy work. I just tell it what to do. What really takes the time is that I have to study the results and make sure they look right and that my instructions aren't at fault somehow. If I were one of those daredevils with utter faith in themselves and the computer, it could all be done in a few minutes."

Pelorat said, "It's really astonishing. Think how much the computer does for us."
"I think of it all the time."
"What would you do without it?"

(Isaac Asimov, Foundation and Earth, 1986)
Abstract

Molecular simulations can access unique atomic-scale information about new materials, pharmaceuticals, and biological environments, making cost-effective predictions and aiding experimental studies. They are particularly useful for describing the mechanisms of nanoscale phenomena and the biological/inorganic interfaces. However, the computational cost of molecular simulations increases with the size of the system as well as with the model complexity, which is related to the accuracy of the simulation. This thesis aims to develop efficient large-scale molecular models that capture important structural details of the atomistic simulations. In particular, we focus on the TiO\textsubscript{2}-lipid interface, which forms in the living cells, exposed to TiO\textsubscript{2} nanomaterials, but is also relevant in the context of biomedical applications. We have studied the interface using atomistic molecular dynamics simulations and found that the characteristics of the lipid adsorption depend on the type of the TiO\textsubscript{2} surface, lipid headgroup composition, and the presence of cholesterol. We then derive a coarse-grained molecular model of the TiO\textsubscript{2}-lipid interface to enable the large-scale simulations of TiO\textsubscript{2} nanoparticles interacting with model cell membranes. We show that the strength of the lipid adsorption increases with the size of the nanoparticle and that a small TiO\textsubscript{2} nanoparticle can become partially wrapped by a lipid membrane. To improve the transferability of the coarse-grained model, we design and test an artificial neural network that learns the interactions in coarse-grained water-methanol solutions from the structural data obtained in multiple reference simulations at atomistic resolution. We show that in the studied system, the neural network learns the many-body interactions and accurately reproduces the structural properties of the solution at different concentrations.
Молекулярное моделирование может предоставить уникальную информацию с атомарным разрешением о новых материалах, лекарственных средствах и биологических средах, что делает его эффективным в прогнозировании и поддержке экспериментальных исследований. Особенной областью применения молекулярного моделирования является исследование процессов, происходящих на поверхности неорганических материалов, которые соприкасаются с биомолекулами. Однако, вычислительная сложность симуляций возрастает с размером моделируемой системы и сложностью модели, которая связана с точностью моделирования. Целью настоящей диссертации является разработка эффективных крупномасштабных молекулярных моделей, воспроизводящих ключевые особенности структуры молекулярных систем, полученных в ходе атомистического моделирования. В этой работе мы сосредотачиваемся на взаимодействии поверхности диоксида титана (TiO$_2$) с молекулами липидов, которое возникает при попадании наночастиц TiO$_2$ в живые клетки организма, что также является актуальным для биомедицинских исследований. Мы изучили взаимодействие с использованием атомистической молекулярной динамики и обнаружили, что адсорбция молекул липидов зависит от типа поверхности TiO$_2$, функциональных групп в полярной части молекул и присутствия холестерина. На основании полученных данных мы разработали грубозернистую молекулярную модель взаимодействия TiO$_2$ с липидами для проведения крупномасштабных симуляций наночастиц диоксида титана, взаимодействующих с липидными мембранами, представляющих упрощённую структуру клеточных мембран. Наши симуляции показывают, что адсорбция липидов растёт вместе с радиусом наночастицы, а наночастица TiO$_2$ с наименьшим радиусом оказывается лишь частично обёрнута липидной мембраной. Чтобы сделать полученные грубозернистые модели более универсальными, мы разработали и отладили нейронную сеть, которая учитывает воспроизводить взаимодействия в грубозернистых водно-метанольных растворах на основании структурных данных, полученных из нескольких атомистических симу-
ляций. Мы демонстрируем, что в данной системе нейронная сеть учитывает многочастичные взаимодействия, что позволяет ей воспроизвести структурные свойства растворов разных концентраций с высокой точностью.
List of Papers

The following papers, referred to in the text by their Roman numerals, are included in this thesis.

PAPER I: *Prediction of Chronic Inflammation for Inhaled Particles: the Impact of Material Cycling and Quarantine in the Lung Epithelium*


M. Ivanov’s contributions: performed molecular dynamics simulations of TiO$_2$ surfaces in contact with lipid molecules, prepared molecular images of the simulated systems, and analyzed the trajectories.

PAPER II: *Atomistic Molecular Dynamics Simulations of Lipids Near TiO$_2$ Nanosurfaces*

M. Ivanov and A. P. Lyubartsev

M. Ivanov’s contributions: ran the simulations, analyzed the data, made the figures, and wrote the manuscript.
PAPER III: Development of a bottom-up coarse-grained model for interactions of lipids with TiO$_2$ nanoparticles
M. Ivanov and A. P. Lyubartsev

M. Ivanov’s contributions: derived and validated the coarse-grained models, ran the large-scale simulations, analyzed the data, made the figures, and wrote the manuscript

PAPER IV: Coarse-Grained Modeling Using Neural Networks Trained on Structural Data
M. Ivanov, M. Posysoev, and A. P. Lyubartsev

M. Ivanov’s contributions: designed the algorithms and implemented them in the software, trained the neural network potentials and ran most of the simulations, analyzed the data, made the figures, and wrote most of the manuscript

Reprints were made with permission from the publishers.
The following is a list of papers by the author not included in this thesis.

PAPER V: **Thermal unwinding of Polyadenylic-Polyuridylic acid complex with TMPyP₄ porphyrin in aqueous solutions**

M. Ivanov, V. Sizov, A. Kudrev


PAPER VI: **Neutron scattering study of polyamorphic THF·17(H₂O) – toward a generalized picture of amorphous states and structures derived from clathrate hydrates**


Abbreviations

AA      All-atom
AAMD    All-atom molecular dynamics
ANN/NN  (Artificial) neural network
CG      Coarse-grained
CGMD    Coarse-grained molecular dynamics
CHL     Cholesterol
DFT     Density functional theory
DMPC    1,2-dimyristoyl-sn-glycero-3-phosphocholine
DPPC    1,2-dipalmitoyl-sn-glycero-3-phosphocholine
HSA     Human serum albumin
IMC     Inverse Monte Carlo
LJ      Lennard-Jones
MC      Monte Carlo
MD      Molecular dynamics
NNP     Neural network potential
NP      Nanoparticle
PBC     Periodic boundary conditions
PME     Particle mesh Ewald
PMF     Potential of mean force
POPE    1-palmitoyl-2-oleoyl-sn-glycero-3-phosphoethanolamine
PPPM    Particle-Particle Particle-Mesh
RDF     Radial distribution function
TEM     Transmission electron microscopy
fs      Femtosecond ($10^{-15}$ second)
ns      Nanosecond ($10^{-9}$ second)
µs      Microsecond ($10^{-6}$ second)
Å       Ångström ($10^{-10}$ meter)
nm      Nanometer ($10^{-9}$ meter)
## Contents

Abstract i  
Sammanfattning ii  
Аннотация iii  
List of Papers v  
Abbreviations ix  
List of Figures xiii  
List of Tables xv  

1 Introduction 17  

2 TiO$_2$ nanomaterials 19  
  2.1 Overview of the nanotoxicity 20  
  2.2 Nanotoxicity mechanisms studies 20  

3 Molecular simulations 23  
  3.1 Molecular dynamics 24  
  3.2 Metropolis Monte Carlo 26  
  3.3 Force fields 29  
    3.3.1 Classical force fields 29  
    3.3.2 Neural network potentials 35  

4 Coarse-grained methods 43  
  4.1 Top-down methods 44  
  4.2 Bottom-up methods 45
4.3 Neural network coarse-graining

5 Summary of the results
5.1 Atomistic simulation studies of phospholipid adsorption on TiO$_2$ surfaces (I, II)
5.2 Development of a bottom-up coarse-grained model for phospholipids and TiO$_2$ surfaces (III)
5.3 Training a coarse-grained neural network potential on structural data (IV)

6 Conclusions and outlook
6.1 Conclusions
6.2 Further developments

Acknowledgments

References
List of Figures

2.1 TEM images of TiO$_2$ NPs ........................................... 19
2.2 TEM images of TiO$_2$ nanotubes with liposomes ............. 21
3.1 Lennard-Jones potential ............................................. 31
3.2 Lennard-Jones fluid in a 2D periodic cell ...................... 34
3.3 A simple fully connected feedforward neural network ...... 36
3.4 Schematic depiction of the Behler-Parrinello NNP approach 42
4.1 Atomistic and coarse-grained representations of a DMPC molecule ..................................................... 44
5.1 Nanomaterial quarantining and cycling ......................... 52
5.2 Formation of a lipid bilayer on TiO$_2$ surface ................. 58
5.3 Lipids adsorbed on anatase (101) surface .................... 53
5.4 Studied lipids ......................................................... 54
5.5 Studied TiO$_2$ surfaces ............................................ 54
5.6 Anatase (101)-POPE number density profiles ............... 58
5.7 Anatase (101)-POPE binding modes residence time histo-
grams ................................................................. 59
5.8 TiO$_2$-lipid CG bead mapping .................................... 62
5.9 Convergence of RDFs during the IMC procedure ............ 63
5.10 Anatase (101)-POPE CG number density .................... 68
5.11 Large anatase slab with POPE lipids ........................... 69
5.12 Anatase NP ($r = 2$ nm)-POPE CG number density ......... 70
5.13 Anatase NP ($r = 10$ nm) and POPE lipids ................... 71
5.14 Anatase NP ($r = 2$ nm) and POPE lipid bilayer ............. 72
5.15 Training CG NNP on structural data ............................ 74
5.16 AA and CG representations of methanol-water mixtures . 79
5.17 LJ NNP model .................................................. 80
5.18 Loss convergence for the CG methanol models .......... 80
5.19 RDF plots for water-methanol mixtures .................... 81
5.20 Angle distribution plots for water-methanol mixtures ... 81
5.21 Three-body potential in CG methanol ........................ 84
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Composition of atomistic systems</td>
<td>55</td>
</tr>
<tr>
<td>5.2</td>
<td>POPE binding modes characteristics</td>
<td>59</td>
</tr>
<tr>
<td>5.3</td>
<td>DMPC binding modes characteristics</td>
<td>60</td>
</tr>
<tr>
<td>5.4</td>
<td>Total fraction of adsorbed lipids for TiO$_2$ nanosurfaces (AA)</td>
<td>60</td>
</tr>
<tr>
<td>5.5</td>
<td>Root-mean-square deviations of headgroup number density</td>
<td>64</td>
</tr>
<tr>
<td>5.6</td>
<td>Total fraction of adsorbed lipids for TiO$_2$ nanosurfaces (CG)</td>
<td>64</td>
</tr>
<tr>
<td>5.7</td>
<td>Composition of large-scale CG systems</td>
<td>65</td>
</tr>
<tr>
<td>5.8</td>
<td>Composition of methanol-water systems in AAMD simulations</td>
<td>78</td>
</tr>
<tr>
<td>5.9</td>
<td>Comparison of the RDF losses</td>
<td>82</td>
</tr>
<tr>
<td>5.10</td>
<td>Comparison of the angle distribution losses</td>
<td>82</td>
</tr>
</tbody>
</table>
1. Introduction

The experimental study of physicochemical phenomena occurring at the interface of two or more distinct phases is significantly complicated by the fact that the typical thickness of an interfacial layer measures not more than 1 nanometer (nm) [1]. At the same time, the interfacial phenomena are ubiquitous in technological applications, geology, the atmosphere, and biological systems to name a few [2]. Molecular simulations can access the atomic scale information that is necessary to study the interfaces, thus complementing advanced experimental techniques [3]. However, the accuracy and predictive power of the simulation methods largely depend on the approximations of the molecular model and specifically how valid is the description of molecular interactions. Additionally, the molecular simulations are always limited by the computing power – increasing the size of the simulated systems or reducing the number of model approximations increases the resource expenditure. Large-scale, or, coarse-grained (CG) molecular models are designed to access larger size and timescales while keeping track only of the most important molecular degrees of freedom [4]. Such a reduction in the number of degrees of freedom, where several atoms can be described as a single interaction site, reduces the computational load significantly. However, the new, effective interactions of the coarse-grained sites have to account for the lost information [5] and it is a challenge to parameterize them accurately [6].

This thesis provides relevant literature and theoretical background and outlines the results of the computational study of interfaces consisting of TiO$_2$ surfaces and nanoparticles with lipid molecules at the atomistic scale (Papers I and II), and the subsequent development of the CG model of the TiO$_2$-lipid interface, based on the atomistic data (Paper III). In addition, Paper IV describes a new coarse-graining method, where the interactions are predicted by a neural network, trained on the structural data from multiple higher-resolution simulations.

The thesis is organized as follows. Chapter 2 gives an overview of the TiO$_2$ nanomaterials, their applications, as well as their adverse health effects in the event of exposure. Particular attention is devoted to the
overview of the studies of the nanotoxicity mechanisms. Chapter 3 is dedicated to the molecular simulation methods – namely the molecular dynamics (MD), Metropolis Monte Carlo (MC), and the classical approximations to describe the molecular interactions. The last part of Chapter 3 outlines the neural network approach to the interactions – the neural network potentials (NNPs). Chapter 4 focuses on different approaches in coarse-grained modeling, including top-down, bottom-up, and neural network coarse-graining methods. Chapter 5 provides a summary of the results, given in Papers I-IV and Chapter 6 – an overall conclusion, and a discussion of possible further developments of this work.
2. TiO$_2$ nanomaterials

Inorganic nanomaterials in the form of nanoparticles, nanotubes, and nanofibers, which measure 1-100 nm in at least one dimension, have seen increased use in a wide range of applications like self-cleaning coatings, solar cells, nanoelectronics, and photocatalysts since the early 2000s [7, 8]. Titanium dioxide (titania) nanoparticles (TiO$_2$ NPs) have become widespread, finding their use in personal care products [9, 10], food [11–13], various paint and self-cleaning coatings [8, 14–17]. Additionally, they play a crucial role in advanced applications like photocatalysts, dyesensitized solar cells, drug delivery, enzyme immobilization and hybrid biomaterials [18–25]. Furthermore, TiO$_2$ NPs are used as a substrate for solid-supported phospholipid bilayers for biosensor applications [26–29]. An illustration of TiO$_2$ NPs is shown in Figure 2.1.

![Figure 2.1: Transmission electron microscopy (TEM) images of TiO$_2$ NPs. A100 – spherical anatase NPs, TiO$_2$ NTs – anatase nanotubes, TiO$_2$ NCs – anatase nanocubes. Provided by experimental collaborators. Reprinted from Paper I.](image)
2.1 Overview of the nanotoxicity

Following the increased abundance of engineered nanomaterials, research studies have raised concerns about the potential health risks associated with TiO$_2$ NPs and other nanomaterials due to their toxicity [12; 30–42]. While macrosized TiO$_2$ is considered biocompatible and non-toxic, toxicological studies have revealed that the uptake of TiO$_2$ NPs can result in adverse effects. For example, it has been shown that compared to microsized titania, TiO$_2$ NPs cause abnormal sedimentation and hemolysis in erythrocyte cultures [36]. Analysis of the damaged cells shows that the attachment of TiO$_2$ NPs changes their surface properties leading to agglutination and cell membrane breakage. Other studies have shown that TiO$_2$ NPs induce cytotoxicity and DNA damage in human amnion epithelial cells [39], as well as phototoxicity and cell membrane damage in keratinocytes under UVA irradiation [38]. Nanotoxicity mechanisms, particularly in the case of metal oxides, involve oxidative stress caused by the generation of reactive oxygen species [37; 40]. Reactive oxygen species have been shown to damage neurons [43] and oxidize and rupture cell membranes [44; 45]. Exposure to TiO$_2$ NPs has also been linked to lung inflammation and increased blood coagulation associated with cardiovascular diseases [33; 34]. Furthermore, it has been suggested that specific interactions of NPs with lipids may result in the so-called quarantining effect, where inhaled NPs accumulate in nanomaterial-lipid aggregates in the extracellular space [33]. TiO$_2$ nanotubes, for instance, show a strong affinity to phosphatidylcholine liposomes as shown in Figure 2.2. Systematic investigations of nanotoxicity emphasize that the toxic effect depends not only on the chemical composition of NPs but also on nanomaterial size, shape, and surface curvature [33; 41; 46]. Despite that many potential adverse effects of TiO$_2$ NPs exposure are known, the molecular mechanisms of the nanotoxicity remain uncertain [47; 48].

2.2 Nanotoxicity mechanisms studies

To study the nanotoxicity mechanisms, numerous experimental studies on model systems have been conducted [29; 35; 44; 49; 55]. It is generally accepted that after entering the organism, the NP attracts proteins, lipids, and other organic molecules to its surface [56]. This organic surface layer is called the biocorona and it defines the further behavior of the NP inside the cell. Coreas et al. [50] have studied the corona formation of TiO$_2$ NPs in simulated gastrointestinal digestion and shown
that lipids dominate the biocorona. In another study, Runa et al. [44] observed that TiO$_2$ NPs attach to the cell surface and oxidize the membrane lipids. Further study has shown that TiO$_2$ NPs can penetrate layers of 1,2-dipalmitoyl-sn-glycero-3-phosphocholine (DPPC) lipids, commonly found in the cell membranes [55]. Yu et al. have provided a more detailed insight into the TiO$_2$-lipid interactions [35]. Their study of TiO$_2$ NPs (20-40 nm in size), formed by different TiO$_2$ polymorphs – anatase and rutile, has found that anatase NPs have a higher affinity towards proteins and impair mitochondrial function. On the other hand, rutile NPs have a higher affinity towards another important cell membrane phospholipid – phosphatidylethanolamine (PE).

While the experimental studies can rarely report on the detailed molecular mechanisms of the nanotoxicity, the molecular simulations offer an atomistic insight where the role of each component can be followed [47, 48, 57, 58]. The simulations can help explain experimentally observed phenomena, such as the chronic inflammation of lung tissues from a single exposure to inhaled NPs [33] as well as structural changes of proteins adsorbed on silver NPs [59, 60]. Not only the simulations are cost-effective [61], but they also provide an alternative to animal testing, often used in experimental nanotoxicity studies [62]. After years of development, computational methods have proved their effectiveness and have been adopted by many regulatory bodies to assess toxicological effects [63]. Independent simulation studies have investigated the lipid adsorption mechanism on TiO$_2$ surfaces, such as the study by Fortunelli and Monti. The authors reported that the adsorption strength is strictly
connected to the nature of both the lipid and the surface [64]. More recently, Schneemilch and Quirke have calculated the adhesion strength of a cell membrane model – 1,2-dimyristoyl-sn-glycero-3-phosphocholine (DMPC) bilayers, to several common TiO$_2$ surfaces [47]. The study suggests that rutile surfaces have a higher affinity towards lipids than anatase surfaces. However, both are lower compared to adhesion to amorphous silica.

Even though the atomistic molecular dynamics can provide an exceptional level of detail in the description of interactions between nanomaterial and biomolecules in aqueous media, the method is currently only viable for simulations of the smallest NPs in a range of 1-5 nm [65, 66]. The sizes of the commonly manufactured nanomaterials, however, usually exceed 10 nm and can go up to 100 nm [42]. To address the computational limitations, coarse-grained (CG) simulations, where the level of detail and the accuracy are sacrificed for the performance, can be used to extend the simulations to larger length and time scales [61, 67]. Lin et al. [57] have simulated the adsorption of a hydrophilic NP on a solid-supported DPPC bilayer using coarse-grained molecular dynamics (CGMD). Their findings suggest that the NP surface charge significantly influences the adsorption. Another CG computational study [48] investigated a small negatively charged NP in contact with a DMPC lipid bilayer and a human serum albumin (HSA) molecule. The simulations have shown that the NP-HSA complex cannot penetrate the lipid bilayer as much as the free NP, which may lead to lower biological activity of the coated NP. A significant number of CGMD studies of NP interactions with lipid membranes have been carried out [68, 77] using Martini force field [78, 80], which is a popular CG molecular model for biomolecular simulations. However, the Martini force field cannot describe a wide array of different nanomaterials in significant detail, limiting the predictive power of such models. The alternatives to the Martini CG approach are discussed in Chapter 4 and the general description of the molecular simulation methods in Chapter 3.
3. Molecular simulations

This chapter describes the basic concepts behind one of the core methods used in this work – molecular simulations. With the advent of computers in the middle of the twentieth century, new methods like Metropolis Monte Carlo (MC) [81, 82] and molecular dynamics (MD) [83–85] were formulated to study complex systems consisting of many interacting particles. These methods allow us to directly sample the positions and momenta of particles, or, the microstates, at the given conditions. Each microstate represents a point in the phase space, that is, in the multidimensional space where each axis is represented by a coordinate or momentum of every particle. The collection of microstates is called a statistical ensemble. The probability of finding a system in a certain microstate corresponds to its representation in the ensemble. Thus, a statistical ensemble can be viewed as a complete representation of a macroscopic system in terms of its microstates and their occurrences. Both molecular dynamics and Metropolis Monte Carlo methods are designed to sample those particular microstates that have the largest statistical weight in the ensemble, thus providing a reasonable estimation of the true ensemble averages that correspond to the observable properties of the system. The simulations allow making predictions about the properties and the behavior of new materials and their reaction to extreme conditions. Moreover, by comparing the predictions and experimental results, one can improve the models and approximations used in the simulations. More often than not, simulations may provide a quicker and cheaper molecular insight than advanced experimental techniques used to study systems at the molecular level. Probing the interfaces proves to be particularly difficult with experiments and, at the same time, rather straightforward using molecular simulation methods.

The first two sections of this chapter are dedicated to the molecular dynamics and Metropolis Monte Carlo methods. The last section describes how the potential energy functions, that is, the interactions between the molecules, may be expressed in the context of classical mechanical (non-quantum) representations of molecular systems, also known as “force fields”.

23
3.1 Molecular dynamics

Molecular dynamics aims to describe the time evolution of a system of interacting particles. If the potential energy functions (Equation 3.1) that govern the interactions between the particles are known, then one can compute the net force acting on every particle in the absence of dissipative forces (Equation 3.2):

\[ U(\mathbf{r}^N) = f(\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_N) \]  

(3.1)

\[ \mathbf{F}_i = -\frac{\partial U(\mathbf{r}^N)}{\partial \mathbf{r}_i} \]  

(3.2)

According to Newton’s laws of motion, the particle’s acceleration is proportional to the net force:

\[ \mathbf{F}_i = m_i \mathbf{a}_i = m_i \frac{d^2 \mathbf{r}_i}{dt^2} \]  

(3.3)

Integrating the equation above yields the information about the particle’s position at any point in time. However, performing analytical integration of the equations of motion for many interacting particles is not feasible. Instead, numerical integration approaches are used in the MD method. In this context, numerical integration means defining a small time step \( \Delta t \) and computing the particles’ positions, velocities, and accelerations only at multiples of \( \Delta t \), one step at a time. Using a large \( \Delta t \) results in significant integration errors and energy drift, whereas, choosing a very small \( \Delta t \) increases the number of time steps required to compute. Ultimately, for the simulation to be stable, the time step should be small enough to capture the fastest motions in the system. For example, in classical atomistic MD, the time step is usually on the order of \( 10^{-15} \) second, or 1 femtosecond (fs). In coarse-grained MD, described in Chapter 4, the time step can be significantly larger, in the 10-100 fs range.

One of the most commonly used numerical integration approaches is the leap-frog algorithm. In a one-dimensional case:

\[ v_{i+1/2} = v_{i-1/2} + a_i \Delta t \]

\[ x_{i+1} = x_i + v_{i+1/2} \Delta t \]

(3.4)
Propagating the equations of motion repeatedly results in the sampling of the phase space of the system, with certain regions being visited more often than others. Those regions contain more probable microstates, which have the larger statistical weight in the final ensemble averages. According to the ergodic hypothesis, a property of the system measured over time converges to the average over the statistical ensemble. Although the hypothesis is often assumed, it is not always valid. Parts of the phase space may become blocked by potential energy barriers such as chemical bonds, molecular rotations, and phase transitions. There exist several methods [86] that help overcome those barriers, allowing a more complete sampling of the phase space, but this topic is out of the scope of this work. If the most important regions of the phase space are visited enough during an MD simulation, the resulting time averages eventually become equivalent to the ensemble averages in the microcanonical (NVE) ensemble, as the number of particles, the volume, and the total energy are conserved during such simulation. Most of the common chemical experiments, however, are performed at constant temperature and pressure. Due to the relation between the average particle velocity and temperature, one can modify the velocities directly to simulate the system’s coupling to a thermostat. In the case of a monoatomic fluid in three dimensions:

\[
\bar{K} = \frac{m \langle v^2 \rangle}{2} = \frac{3}{2} k_B T
\]

\[
T = \frac{m \langle v^2 \rangle}{3k_B}
\]

where \( \bar{K} \) is the average kinetic energy per particle, \( \langle v^2 \rangle \) is mean squared velocity and \( k_B \) is the Boltzmann constant. The simplest way to enforce a certain temperature on the system would be to multiply all of the velocities by the following scaling factor at every time step:

\[
\alpha = \sqrt{\frac{\bar{K}}{K}}
\]

where \( \bar{K} \) is the average kinetic energy corresponding to the target temperature and \( K \) is the current kinetic energy of the system. The problem with this approach is that even though the temperature stays the same, due to the absence of fluctuations in the resulting kinetic energy, the sampled microstates do not correspond to the canonical (NVT) ensemble. The Bussi (v-rescale) thermostat algorithm [87] solves this problem by using the kinetic energy \( K \) that is instead drawn from the canonical
equilibrium distribution at the desired temperature. To obtain the correct kinetic energy, one propagates it according to the following equation:

\[ dK = (\bar{K} - K) \frac{dt}{\tau} + 2 \sqrt{\frac{\bar{K} K}{N_f}} \frac{dW}{\sqrt{\tau}} \]  

(3.7)

where \( \tau \) is the characteristic thermostat coupling time scale, \( N_f \) is the number of degrees of freedom and \( dW \) is a Wiener noise. Other examples of thermostat algorithms are Berendsen [88], Andersen [89], and Nose-Hoover [90; 91] thermostats. Similarly to the temperature coupling, one can rescale the coordinates of the particles and keep the pressure constant during the simulation. Examples of such barostat algorithms are Berendsen [88], Parrinello-Rahman [92; 93], and stochastic cell rescale [94] barostats.

Using both thermostat and barostat algorithms during an MD simulation may then result in the sampling of the isothermal-isobaric (NpT) ensemble, which makes it more straightforward to compare the results with the experimental data.

3.2 Metropolis Monte Carlo

The Monte Carlo method, in general, is a large array of different numerical methods that rely on repeated random sampling, and the method finds its applications far beyond the molecular modeling domain. One of the most basic and well-known applications of the Monte Carlo method is the estimation of the number \( \pi \). It is done by sampling a large number of pairs of random numbers, uniformly distributed in the \([0, 1]\) range, which represent the \(x\) and \(y\) coordinates of points on a unit square. By computing the ratio of points residing within a unit radius to the total number of sampled points, one can get an estimate of the area of a unit quarter circle. Multiplying it by a factor of four yields an estimate for \( \pi \). Although simple, this method of estimating \( \pi \) is significantly less efficient than more conventional numerical integration techniques that rely on evaluating the integrand on a set of predefined points and summing up the results. However, as the problem’s dimensionality increases, the Monte Carlo approach becomes an integration method of choice, as its error does not depend on the number of dimensions. In contrast, the number of integration points in the deterministic methods increases exponentially.

In the context of molecular modeling, one can formulate the prob-
lemma of finding the configuration-dependent ensemble averages $\langle A \rangle$ (like density, radial distribution functions, etc.) as a random variable with a certain probability density function. In the canonical (NVT) ensemble:

$$\rho (r^N) = \frac{1}{Z} \exp [-\beta U(r^N)]$$

(3.8)

where $\rho (r^N)$ is the probability density function, $\beta$ is the inverse temperature or $1/k_B T$, $U(r^N)$ is the potential, and $Z$ is the configurational part of the partition function, or, the configurational integral, that ensures the normalization of the probability density function (constant terms in front of the integral are ignored):

$$Z = \int d r^N \exp [-\beta U(r^N)]$$

(3.9)

where the integration is done over the whole configurational part of the phase space. Finally, an expression for the ensemble average $\langle A \rangle$:

$$\langle A \rangle = \int d r^N A(r^N) \rho (r^N) = \frac{1}{Z} \int d r^N A(r^N) \exp [-\beta U(r^N)]$$

(3.10)

What makes this integration difficult is not only that it is $3N$-dimensional (where $N$ is the number of particles) but also the fact that the Boltzmann factor $\exp [-\beta U(r^N)]$ in the integrand makes it highly non-uniform. A tiny portion of the phase space contributes the most to the ensemble averages, while the rest has vanishingly small statistical weight due to the exponential nature of the Boltzmann factor and a strong variation of the potential energy with particles’ positions. This makes it impossible to find such ensemble averages using a standard Monte Carlo approach. Metropolis Monte Carlo method, however, solves this problem by sampling the configurations according to the probability density $\rho (r^N)$ (Equation 3.8). The ensemble average $\langle A \rangle$ then can be estimated as a plain arithmetic average:

$$\langle A \rangle \approx \frac{1}{M} \sum_{i=1}^{M} A(r^N_i)$$

(3.11)

where $M$ is the number of configurations $r^N$ generated according to the probability distribution $\rho (r^N)$. A basic algorithm for generating such configurations is as follows. Starting from some initial configuration $r^N_0$, we generate a new state $r^N_1$ by making a small random change to
the previous state. We denote the probability of a specific transition from configuration \( r^N_0 \) to \( r^N_1 \) as \( \alpha(r^N_0 \rightarrow r^N_1) \). The most common ways of generating a new configuration are a random displacement of a single atom, a whole molecule, or a molecular rotation. After the new state is generated, we accept it with a certain probability \( acc(r^N_0 \rightarrow r^N_1) \). The total transition probability \( \pi(r^N_0 \rightarrow r^N_1) \) is then a product of the two probabilities:

\[
\pi(r^N_0 \rightarrow r^N_1) = \alpha(r^N_0 \rightarrow r^N_1) \times acc(r^N_0 \rightarrow r^N_1) \tag{3.12}
\]

If the state is accepted, we continue making changes and generate new states. Otherwise, we reject the new state and perform another random change to the original state. This procedure is called a Monte Carlo step. At any point during the Monte Carlo procedure, whether the step was accepted or rejected, any property \( A \) that depends on the configuration can be computed and stored for later averaging. By enforcing the condition of detailed balance on the transition probabilities, one ensures that after generating a large number of configurations \( r^N_i \), the probability of finding a system in a state \( r^N_i \) is equal to \( \rho(r^N_i) \):

\[
\rho(r^N_i) \pi(r^N_i \rightarrow r^N_j) = \rho(r^N_j) \pi(r^N_j \rightarrow r^N_i) \tag{3.13}
\]

If the way the new states were generated is symmetrical (i.e. there is the same probability of going forward and backward), then we can compute the relation of transition probabilities as:

\[
\frac{\pi(r^N_i \rightarrow r^N_j)}{\pi(r^N_j \rightarrow r^N_i)} = \frac{\rho(r^N_j)}{\rho(r^N_i)} = \exp \left[ -\beta (U(r^N_j) - U(r^N_i)) \right] \tag{3.14}
\]

There are different ways to satisfy the condition above, but the one that was used in the original work by Metropolis et al. \[82\] is to accept any new configuration that has lower energy \( U \) and accept configurations with higher energy \( U \) with probability \( \exp(-\beta \Delta U) \).

For ergodic systems in the limit of a large number of configurations, the Metropolis Monte Carlo method generates statistical averages that are equivalent to the time averages from MD. However, the efficiency of each method in terms of computation resources spent to generate results of a certain quality differs and depends on the specific system and the conditions. For example, in systems with a relatively low particle density, larger Monte Carlo steps can be performed which improves the sampling efficiency. Molecular dynamics, on the other hand, does not get any specific speed up when the particle density is reduced, but the
number of particles stays the same. Another advantage of the Metropolis Monte Carlo method is that each CPU core can perform its independent simulation, and the results from each core can be gathered in the end, minimizing the parallelization overhead.

3.3 Force fields

3.3.1 Classical force fields

Both molecular dynamics and Metropolis Monte Carlo methods rely on how accurate is the representation of the potential energy in the system. One of the most accurate ways to describe the interactions in molecular systems is to account for both nuclear and electronic degrees of freedom with the help of quantum chemistry, or *ab initio* methods. However, such methods face several difficulties such as an extremely high computational complexity and non-linear scaling. Currently, it becomes unfeasible to perform an MD simulation where the forces are computed *ab initio* if the number of atoms exceeds a few thousand atoms. For this reason, more simple and approximate ways to describe the interactions are used to perform large-scale molecular simulations. In particular, various classical approximations are used to describe interactions, which means that the electronic degrees of freedom are accounted for implicitly. Within the classical approach, the atoms are most often described as mass points and are assigned a fixed electric charge, representing not only the balance between the number of electrons and protons in the atom but also its relative electron density within a molecule. The non-electrostatic part of the potential energy can be represented in a number of ways. First of all, the interactions are usually separated into bonded and non-bonded contributions. With the bonded interactions one can describe covalent bonds, bond angles, as well as dihedral angles. The covalent bonds and bond angles are most often described as classical harmonic oscillators, which can describe the position at the equilibrium very well:

\[
U(r_{ij}) = K_b(r_{ij} - r_0)^2 \\
U(\theta_{ijk}) = K_\theta(\theta_{ijk} - \theta_0)^2
\]

(3.15)

where \( r_{ij} \) and \( r_0 \) are the current and equilibrium distances, \( \theta_{ijk} \) and \( \theta_0 \) are the current and equilibrium bond angles, and \( K_b \) and \( K_\theta \) represent force constants, which define the stiffness of the oscillator. It might be beneficial, however, to constrain the bond length completely, particularly in the case of fast oscillations in MD simulations, like in covalent bonds.
with a hydrogen atom. Bond constraints remove the fastest degrees of freedom, thus allowing to increase the time step. There are several constraint algorithms, most notably SHAKE [96], SETTLE [97] and LINCS [98][99]. To correctly describe intramolecular rotations and the flexibility of long molecular chains, one uses an additional dihedral potential:

$$U(\phi_{ijkl}) = K_\phi \left[ 1 + \cos (n\phi_{ijkl} - \delta) \right]$$  \hspace{1cm} (3.16)

where $K_\phi$ is the dihedral force constant, $n$ is the multiplicity of the potential, $\phi_{ijkl}$ is the dihedral angle and $\delta$ is the phase shift. Usually, all atomic pairs that are not connected through the bonds or angles interact through the non-bonded potential. As was mentioned before, within the classical molecular models, each atom is usually assigned a fixed charge. The electrostatic interactions are then computed according to the Coulomb's law:

$$U(r_{ij}) = \frac{1}{4\pi\varepsilon_0} \frac{q_i q_j}{r_{ij}}$$  \hspace{1cm} (3.17)

where $r_{ij}$ is the distance between atom $i$ and atom $j$, $\varepsilon$ is the relative dielectric permittivity, $\varepsilon_0$ is the dielectric permittivity of vacuum and $q_i$ and $q_j$ are the charges on both atoms. In addition to the electrostatic interactions, both charged and neutral atoms attract each other through the dispersion interactions that originate from the mutually induced dipoles on the interacting atoms. It has been shown that the strength of those interactions is proportional to $r^{-6}$. At very short distances, the atoms strongly repel each other due to the Pauli expulsion. This repulsion term is best described by an exponential term $\exp(-\alpha r)$. However, to simplify the computations, instead of computing the exponent, one usually takes a square of $r^{-6}$ that is used to compute the dispersion term to get $r^{-12}$, which approximates the exponential repulsion reasonably well. A combination of the dispersion and repulsion terms is expressed in a well-known Lennard-Jones (LJ) potential (Equation 3.18) and an example of such potential is shown in Figure 3.1:

$$U(r_{ij}) = 4\varepsilon_{ij} \left[ \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left( \frac{\sigma_{ij}}{r_{ij}} \right)^6 \right]$$  \hspace{1cm} (3.18)

where $\varepsilon_{ij}$ and $\sigma_{ij}$ are the so-called LJ parameters that are, in principle, unique to atomic pair $ij$. However, in practice, those parameters are usually assigned to each atomic type only, which is an atom in a specific chemical environment (e.g., $sp^3$ or $sp^2$ carbon, oxygen in a water molecule
or a carboxyl group, etc.). Then, to compute LJ potential for a pair of atoms, one uses combination rules, for example, Lorentz-Berthelot rules [100]:

\[
\begin{align*}
\sigma_{ij} &= \frac{\sigma_i + \sigma_j}{2} \\
\epsilon_{ij} &= \sqrt{\epsilon_i \epsilon_j}
\end{align*}
\] (3.19)

where \(\sigma_{ij}\) and \(\epsilon_{ij}\) are the combined LJ parameters computed from atom-specific parameters. The physical meaning of both LJ parameters can be deduced from Figure 3.1. The parameter \(\sigma\), which has a unit of distance, controls the effective radius of the atom. At \(r = \sigma\), the strength of attractive and repulsive interactions is balanced. The parameter \(\epsilon\) has a unit of energy and sets the depth of the potential well. At the distance \(\sqrt{2} \sigma\), the overall interaction has the lowest energy and equals \(-\epsilon\).

The total potential energy of a molecular system, described by such
classical approximation, would be a combination of the terms presented above and can be written as follows \[101\]:

\[
U(r^N) = \sum_{bonds} K_b(r_{ij} - r_0)^2 + \sum_{angles} K_\theta(\theta_{ijk} - \theta_0)^2 \\
+ \sum_{dihedrals} K_\phi \left[ 1 + \cos(n\phi_{ijkl} - \delta) \right] \\
+ \sum_{i \neq j} 4\varepsilon_{ij} \left[ \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left( \frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] + \sum_{i \neq j} \frac{1}{4\pi\varepsilon_0} \frac{q_i q_j}{r_{ij}}
\]

Equation 3.20

One can notice from the equation above that for this classical approximation to work, one needs to derive a significant number of parameters that are unique to every covalent bond, angle, dihedral angle, and atom type. In addition, there are certain requirements for these sets of parameters – first of all, they should describe the molecular interactions reasonably well and reproduce at least some of the known properties of the molecular system. Moreover, these sets of parameters should be transferable to some extent, which means that one can use them for different systems, varying conditions, and compositions and expect reasonable results. A set of such parameters that are established and widely used in molecular modeling communities are usually referred to as the force fields. A few examples of general force fields that are designed to simulate a wide range of molecular species are GAFF \[102\], OPLS-AA \[103\], UFF \[104\], GROMOS \[105\] and CGenFF \[106\]. Other force fields can be more specialized, like, CHARMM36 \[107\] and AMBER ff14SB \[108\] force fields for proteins, lipids, and nucleic acids, Lipid14 \[109\] and Slipids \[101; 110; 111\] force fields for lipid molecules. All of the force fields mentioned above use very similar functional forms to describe the potential energy (with some minor differences and additional secondary terms) but differ in the way the parameters are obtained and their exact values, as well as in the scope of the force field itself. To derive the force field parameters, one usually uses a combination of data from \textit{ab initio} simulations as well as fitting to experimental data.

It should be noted that such force fields have a number of limitations. Firstly, only the pair contributions are considered in the non-bonded part of the potential, even though in certain systems, like water-ion clusters, the many-body interactions have a considerable effect \[112\]. Moreover, the potential energy formulation used in Equation 3.20 does not include explicit polarizability, which affects interfacial properties \[113\]. There are examples of polarizable force fields, however, such as the CHARMM
polarizable force field [114]. Finally, the force fields with a functional form similar to Equation 3.20 have fixed covalent bonds, which prohibits the formation of new molecular species in the simulation. A notable example of a force field that supports bond formation and dissociation is ReaxFF [115]. However, a common limitation for both reactive force fields and polarizable force fields is that they use more potential energy terms, have to account for more degrees of freedom, and a significantly shorter time step has to be used.

The approximated description of the molecular systems, enabled by the classical force fields, allows a significant reduction of complexity, in comparison to the quantum mechanical description. However, currently, the state-of-the-art methods reach not more than a few billion atoms in an MD simulation using such classical force fields. Although it is significantly more than within the quantum mechanical description, it is still extremely far from the macroscopic amount of matter, which contains on the order of $N_A$ number of molecules (Avogadro constant), or $\sim 6.022 \times 10^{23}$. For example, a simulation of a million water molecules would not provide an accurate description of a macroscopic amount of water, but rather of a nanodroplet of water, which has different properties because a high proportion of the molecules is located on the surface. However, this problem of having too many surface molecules that are different from those in the bulk phase can be solved by imposing periodic boundary conditions (PBC) on the simulated system. An illustration of two-dimensional (2D) PBC is shown in Figure 3.2. The main idea behind this approach is to place the simulated molecular system in a so-called simulation cell, or simulation box with a defined size, and allow the molecules that are close to a cell boundary to both move to and interact with the portion of the box on the opposite side of the boundary. This can be interpreted as filling the whole space around the simulation box with an infinite number of its copies. Cubic simulation boxes are a popular choice, but other shapes that can fill the space entirely, such as truncated octahedrons and rhombic dodecahedrons are used as well. One way of implementing the PBC is to allow every particle to interact only with the nearest image of another particle, whether it is located in the simulation box or one of the periodic images outside of the simulation box. Additionally, whenever a particle crosses the periodic boundary, its coordinates are often adjusted so it appears on the other side of the simulation box. Another way of accounting for the PBC is to compute the interactions for each particle only within a certain cut-off radius, meaning that the particles that are further than a certain distance do not contribute to the potential energy of the particle. This cut-off sphere
includes both the particles from the main simulation box but also from the neighboring periodic images for the PBC to work correctly.

Figure 3.2: Lennard-Jones fluid in a 2D periodic cell. The atoms belonging to the simulation box are in the center and are shown as opaque spheres enclosed by the solid blue frame. The transparent spheres enclosed by the dashed blue frames represent the copies (periodic images) of the original system.

By effectively surrounding the simulated system with its copies, one can get rid of the surface effects entirely. Instead, one simulates a bulk periodic system, and if the simulation cell is sufficiently large, one can reduce the effect of periodic artifacts. This way it is possible to obtain the correct bulk properties of liquids with as little as a few thousand atoms in the simulation box. The majority of molecular simulations of liquids, macromolecules, and biomolecular systems, including the present work, are performed with the use of such three-dimensional PBCs. It should be noted, however, that using a plain cut-off for computing interactions and disregarding any non-bonded pairs that are further than a certain dis-
tance may lead to severe artifacts. First of all, truncating interactions at the cut-off distance may result in discontinuities in the potential, but this problem can be solved by introducing a constant shift to the potential so that it becomes zero at the cut-off. While this works quite well for LJ interactions that are not very long-ranged, the electrostatic potential does not decay as quickly, resulting in significant long-range contributions that cannot be accounted for correctly just by increasing a cut-off radius in a system with PBC. Instead, one computes the long-range electrostatic potential coming from an infinite number of periodic images using such methods as Ewald summation [116], Particle mesh Ewald (PME) [117, 118] and particle-particle particle-mesh (PPPM or P3M) [119]. All three methods are based on performing a Fourier transform and summing up the long-range contributions in the reciprocal space. In PME, for example, the simulation box is separated by a grid and each grid point is assigned a charge from its surroundings. Then, the reciprocal energy is computed by a 3D Fast Fourier Transform (FFT) of the grid and the potential is recovered by inverse transformation. Similarly, one can apply the PME approach to compute the long-range LJ interactions [120] as the analytical solution is only available for isotropic systems.

3.3.2 Neural network potentials

As an alternative to the theoretical expressions used in classical force fields, one can train and use artificial neural networks (ANN or NN) to compute the potential energy in a system of interacting atoms. The artificial neural network can be understood as a framework for expressing a relationship between different sets of data, in other words – a mathematical function. One of the main features of a neural network is the use of a large number of adjustable parameters called weights and biases, which in combination with the given input data provide the desired output data, or result. In one of the simplest NN architectures, the fully connected feedforward neural network, the numerical input data is arranged in the input layer of the NN, where each number is placed in a neuron. Then, the input data is fed to a sequence of interconnected layers that are called the hidden layers, each consisting of a different number of neurons, before finally reaching the output layer. The output layer, also consisting of any number of neurons, contains the final result that we want to obtain from the given input data. Every connection between two neurons is assigned a unique weight parameter and every hidden and output neuron – a bias parameter. An illustration of such a neural network is shown in Figure 3.3. The value of a neuron in a hidden
or an output layer is computed from the data in the previous layer in the following way:

\[ n^k_l = f_a \left[ \sum_i \left( w^{k-1}_{il} \times n^{k-1}_l \right) + b^k_l \right] \]  

(3.21)

where \( n^k_l \) represents the value of a neuron \( l \) in the layer \( k \), \( w^{k-1}_{il} \) expresses the weight parameter that is assigned to a connection between the neuron \( n^k_l \) and a neuron \( i \) in the previous layer \( k-1 \), or \( n^{k-1}_l \). \( b^k_l \) shows the bias parameter of a neuron \( n^k_l \) and \( f_a \) is the activation function. Activation functions are used to introduce non-linear behavior to the NNs. A few examples of such activation functions are the sigmoid: \( f(x) = \frac{1}{1+e^{-x}} \), hyperbolic tangent: \( f(x) = \tanh(x) \) and the rectified linear unit or ReLU: \( f(x) = \max(0,x) \).

**Figure 3.3:** A simple fully connected feedforward neural network with two input neurons \( X_1 \) and \( X_2 \), three hidden neurons \( n_1 \), \( n_2 \) and \( n_3 \) and a single output neuron \( Y_1 \) organized in three layers. \( w^{kn}_lm \) and \( b^{kn}_l \) are the weight and bias parameters where \( k \) denotes the layer index, \( l \) – the neuron index in the current layer and \( m \) – the neuron index in the next layer.
To get the correct results from the given inputs, one has to tune the weights and biases, or, in other words – train the neural network. NN learns by example – one provides the NN with a large number of input-output pairs, and the NN, starting from some initial guess for the parameters (usually random numbers from -1 to 1 for weights and 0 for biases), attempts to predict the known output and correct its parameters until the prediction error is minimized. The method that is used to train the NN (by correcting the weights and biases) is called backpropagation \[121\]. To perform backpropagation, one introduces the loss function, which in some way reflects how erroneous is the prediction with the given set of the NN parameters. Often, one uses the mean squared error (MSE) function:

\[
L = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{3.22}
\]

where the \(y_i\) is the true value of the output, \(\hat{y}_i\) is the corresponding prediction given by the neural network and \(n\) is the number of the output neurons. The core idea of the backpropagation is to minimize this loss function \(L\) by computing the partial derivatives of the loss function with respect to weights and biases – \(\frac{\partial L}{\partial w_i}\) and \(\frac{\partial L}{\partial b_i}\). Although the number of derivatives equals the number of parameters in the neural network, they are relatively straightforward to compute since every derivative can be computed analytically. If the derivatives are known, one can update the parameters using such methods as the steepest descent:

\[
w_i' = w_i - \eta \frac{\partial L}{\partial w_i} \tag{3.23}
\]

where \(w_i'\) is the updated value of the weight parameter \(w_i\) and \(\eta\) is the learning rate, which controls the step size of the gradient descent. Such optimization does not guarantee that the global minimum of the loss function can be found, however, and more advanced optimizer algorithms exist that improve the convergence of the loss function. More details on the NN architectures, learning, and optimizers can be found in this online book \[122\]. Once the loss function is minimized and the NN is trained, it can be used to make predictions for a new set of inputs for which the output is not yet known. Often, one uses a fraction of the whole set of known inputs and corresponding outputs, to test the validity of the NN, before letting the NN make predictions for the new data. It has been proven that the feedforward neural networks, by virtue of having many hidden layers, many adjustable parameters as well as non-linear
activation functions, in principle, can fit any possible multidimensional real-valued function with arbitrary precision, making them an extremely powerful tool \cite{123-126}. Increasing the number of NN parameters and the amount of training data makes them more and more flexible, allowing them to solve specialized tasks with unparalleled efficiency.

The predecessors of the modern artificial neural networks appeared in the middle of the twentieth century and have been used to solve various tasks such as making predictions, classifications, and parametrizations \cite{127-131}. However, at that time, the neural networks could not compete with the traditional approaches and did not gain much attention. Since the early 2000s, thanks to the improved NN algorithms and increased computing power, in particular in the form of graphics processing units (GPUs), the application of NNs has become widespread in computer vision, forecasting, natural language processing, and lately in text and image generation \cite{132-134}. The neural networks have caught the attention of molecular modeling researchers as well, and have found their uses in \textit{ab initio} quality neural network potentials, coarse-grained molecular dynamics, free energy methods, sampling of molecular equilibrium structures and drug design \cite{135; 136}. This section focuses on the neural network potentials (NNPs) in particular.

The NNP can be viewed as an alternative way to express the potential energy function using a neural network. The NNP accepts the atomic coordinates as inputs and outputs the potential energy. In most NNP designs, the training data for such a potential consists of a large number of different atomic configurations and corresponding potential energy (or forces) computed with an \textit{ab initio} method \cite{137}. An obvious disadvantage of such NNPs is the need for training as well as the relative lack of transferability – the predictive power of the NNPs is limited by the training data. However, once the NNP is trained it can produce accurate results for a very low cost. One of the earliest works on the NNPs was published in the 90s by Doren and co-workers \cite{138}, where the adsorption of hydrogen on a silicon surface was studied with an NNP trained on DFT data. The advantages of this approach were evident immediately – the NN potential could make chemically accurate predictions for systems with several degrees of freedom for a fraction of the cost of the reference, \textit{ab initio} simulation. However, the early NNPs faced several problems. First of all, most of them were trained to predict the total potential energy of a system, which prohibited them from simulating a variable number of atoms as the size of the input layer is fixed. Additionally, there was a lack of generally applicable input coordinates that were invariant to translations, rotations, and permutations of iden-
tical atoms, thus, the NNPs were mostly confined to the systems with a relatively small number of degrees of freedom \([126]\). In 2007, Behler and Parrinello \([139; 140]\) proposed two major changes to the NNP approach. To facilitate the simulation of more varied systems with the number of atoms that is different from the reference data, they trained the NNP to predict an energy contribution \(E_i\) to the total potential energy \(E\) from a single atom \(i\), instead of training the NNP to predict the potential energy of the whole system. The total potential energy is computed as a sum of all energy contributions:

\[
E = \sum_i E_i
\]  

(3.24)

Each chemical element (carbon, oxygen, hydrogen, etc.) gets an NNP with a unique set of weights and biases, which helps differentiate the elements. Thus, the NNPs within this approach are often referred to as atomic NNPs. With such atomic NNPs, one can add many different atoms (for which the atomic NNPs are available) to the simulated system and still be able to evaluate the total potential energy. A fixed set of weights and biases in each atomic NNP ensures the permutation invariance for identical atoms. As in the other NNPs, the Behler-Parrinello NNPs learn by reproducing energies of a large number of atomic configurations with known potential energies obtained with an accurate quantum chemical computation method. Another major difference is the use of a new type of atomic descriptor – the atom-centered symmetry function, which serves as an input to an atomic NNP. Usually, a combination of several different symmetry functions is used. The symmetry function provides information about the local environment of an atom, including the positions of neighboring atoms, which is relevant for energy prediction. The simplest symmetry function, proposed by Behler \([140]\) and denoted as \(G^1\), simply counts the number of atoms within a certain cut-off radius:

\[
G^1_i = \sum_{i \neq j} f_c(r_{ij})
\]  

(3.25)

where \(G^1_i\) is the \(G^1\) symmetry function of an atom \(i\), \(r_{ij}\) is the Euclidian distance from the atom \(i\) to an atom \(j\) and \(f_c\) is the smooth cut-off function:
\[ f_c(R_{ij}) = \begin{cases} 
0.5 \times \left[ \cos \left( \frac{\pi r_{ij}}{r_C} \right) + 1 \right] & \text{for } r_{ij} \leq r_c. \\
0 & \text{for } r_{ij} > r_c. 
\end{cases} \] (3.26)

where \( r_c \) is the cut-off radius. A more advanced type of symmetry function called the radial symmetry function or \( G_2 \) is shown here:

\[ G_i^2 = \sum_{i \neq j} \exp \left[ -\eta (r_{ij} - r_s)^2 \right] \times f_c(r_{ij}) \] (3.27)

where \( G_i^2 \) is the \( G^2 \) symmetry function of an atom \( i \), \( \eta \) parameter sets the width of the Gaussian curve and \( r_s \) defines its center. Another important type of symmetry function is the angular symmetry function or \( G_3^3 \):

\[ G_i^3 = 2^{1-\zeta} \sum_{i \neq j \neq k} (1 + \lambda \cos \theta_{ijk})^\zeta \times \exp \left[ -\eta (((r_{ij} - r_s)^2 + (r_{ik} - r_s)^2 + (r_{jk} - r_s)^2) \right] 
\times f_c(r_{ij})f_c(r_{ik})f_c(r_{jk}) \] (3.28)

where \( \theta_{ijk} \) is an angle between the triplet \( ijk \), \( \zeta \) sets the angular resolution and the parameter \( \lambda \) may take values of \(+1\) or \(-1\), shifting the maxima of the cosine function to \( \theta_{ijk} = 0^\circ \) and \( \theta_{ijk} = 180^\circ \). A combination of \( G^1 \), \( G^2 \), and \( G^3 \) symmetry functions with different \( \eta \), \( r_s \), \( \lambda \), and \( \zeta \) parameters, tuned for specific atomic pairs and triplets, can be used to fully describe the local environment of an atom. The symmetry functions depend on the coordinates of multiple neighboring atoms within a cut-off sphere and thus they can be considered many-body descriptors, encoding many-body interaction terms. Moreover, they are invariant to translations and rotations of the whole system, which makes them a great candidate for an atomic descriptor in NNPs. Figure 3.4 shows a diagram representing the Behler-Parrinello approach. Such NNP allows computing the total potential energy of a system composed of a large number of different atoms in any configuration with an \textit{ab initio} accuracy, including the many-body terms, at a significantly lower computational cost [126]. With such an ability, one can run a Metropolis Monte Carlo simulation and sample configurational averages. However, computing the forces acting on each atom and running an MD simulation is also possible, since the total potential energy is an analytical function of the Cartesian coordinates:
\[ \vec{F}_k = -\frac{\partial E}{\partial \vec{r}_k} = -\sum_i^N \frac{\partial E_i}{\partial \vec{r}_k} = -\sum_i^N \sum_j^M \frac{\partial E_i}{\partial G_{i,j}} \frac{\partial G_{i,j}}{\partial \vec{r}_k} \] (3.29)

where \( \vec{F}_k \) is the force acting on an atom \( k \) located at \( \vec{r}_k \) in a system of \( N \) interacting atoms, each described by \( M \) symmetry functions. It becomes evident that the smoothness of the symmetry functions is crucial to obtain continuous forces. This NNP method is implemented in n2p2 library [141] within the LAMMPS molecular dynamics simulation software [142] and has been utilized to simulate a wide range of different systems like elementary silicon, carbon, metal oxides, water clusters, water-metal oxide interfaces as well as reactive organic mixtures containing carbon, oxygen, hydrogen and nitrogen [126; 139; 143; 144]. Other approaches to NNPs and molecular simulations using neural networks include but are not limited to the Gaussian Approximation Potentials (GAP) [145–149], the electrostatic embedding scheme [150], Deep potential for molecular dynamics (DeePMD) [151; 152], graph neural networks approaches [153; 154], gradient-domain machine learning (GDML) [155; 156] and hybrid force field-neural network approaches [157]. More information can be found in a recent review paper [158]. However, the ideas from the Behler-Parrinello NNP approach in particular were taken as a basis for the coarse-grained NNP developed in this work.
Figure 3.4: Schematic depiction of the Behler-Parrinello NNP approach for a system containing $N$ interacting atoms of a single kind. A set of symmetry functions is computed for each atom from its coordinates and the coordinates of the neighboring atoms. The symmetry functions $\{G_i\}$ are fed to the atomic NNP that outputs the energy contribution. The contributions from every atom are summed into the total potential energy. Since there is only one kind of atom in the system, the weights and biases in the atomic NNP are the same. Otherwise, there are as many unique atomic NNPs as different chemical elements.
4. Coarse-grained methods

Compared to the classical atomistic force fields described in the previous chapter, the coarse-grained (CG) approach offers even more computational resource savings, especially when simulating molecular systems at larger scales, where the simulation box sizes can reach hundreds of nanometers (nm) [159]. This can be achieved by systematically reducing the total number of degrees of freedom in the simulated system, leaving only the most important ones. Usually, specific groups of atoms are merged into a single coarse-grained particle or a CG bead, reducing the number of interaction sites and thus the computational load. An example of such grouping can be illustrated on a DMPC phospholipid molecule (1,2-dimyristoyl-sn-glycero-3-phosphocholine) in Figure 4.1. Such a CG representation of a DMPC molecule reduces the number of interaction sites from 118 atoms in the atomistic representation to only 10 CG sites. At the same time, many important chemical features of the molecules are still captured at the CG resolution – a polar headgroup with a positively charged choline group and a negatively charged phosphate group, as well as two non-polar hydrocarbon tails. In addition to the reduced number of CG sites compared to the number of atoms at the atomistic resolution, often the solvent is described implicitly, meaning that no interaction sites belonging to the solvent are present in the system and the effect of the solvent is included into the interactions of the remaining CG beads [4]. Furthermore, in coarse-grained molecular dynamics simulations (CGMD), the time step can be significantly increased in comparison to the atomistic MD, as the new degrees of freedom are not as fast as at atomistic resolution.

In summary, the CG methods provide the speed-up compared to the molecular simulations at an atomistic resolution based on three different factors: 1) reduced number of total interaction sites; 2) increased time step in CGMD simulations; 3) faster configurational sampling due to reduced friction, especially in the absence of explicit solvent [79; 160]. The last factor becomes problematic if the dynamical properties are of interest, however, it does not interfere with the configurational averages. Moreover, the correct dynamics can be recovered through Brownian dy-
namics which simulates the collisions with the solvent molecules through the addition of random forces on the CG beads as well as the inclusion of the solvent viscosity [161] [162].

Figure 4.1: Atomistic (left) and coarse-grained (right) representations of a DMPC molecule. Adapted from Paper III.

4.1 Top-down methods

An evident problem of the CG method is that the new, coarser representation of a molecular system now requires a different way of expressing the interactions of the CG beads. One way of defining them is to parameterize a whole new force field for the CG representation to reproduce experimentally measurable macroscopic quantities of the system, in a so-called top-down approach. In biomolecular modeling, the MARTINI force field [79] [163] is one of the most notable examples. Each type of CG bead in the MARTINI force field represents a “molecular building block”, usually composed of four heavy atoms (e.g. carbon, oxygen, nitrogen, but not hydrogen), used to construct different biologically relevant organic molecules, such as phospholipids, proteins, and sugars, but also
other organic molecules, water, and even nanoparticles. The non-bonded interactions are described by the Lennard-Jones potential in combination with Coulomb interactions, as the polar CG beads have an electric charge. The harmonic potential is used to simulate the bond stretching and the angle bending potentials. The non-bonded part is calibrated with experimental thermodynamic data, most notably the partitioning free energies and the miscibility data, and the bonded potentials are parameterized to reproduce the atomistic structure of a molecule. After more than two decades of development, the MARTINI force field became a truly general CG force field, encompassing a wide range of biological and small organic molecules, as well as materials, such as polymers, surfaces, and nanoparticles [164–166]. However, when using the MARTINI force field one can expect incorrect phase transition temperatures and nonquantitative agreement with experimental free energies of solvation due to the use of non-specific LJ potential and an entropy-enthalpy imbalance caused by the reduced number of degrees of freedom and thus the entropy in the CG systems [166]. Furthermore, the entropy-enthalpy imbalance leads to an incorrect description of the temperature dependence of the hydrophobic effect, adversely affecting the force field transferability with respect to temperature [165]. Parameterization of adsorption of organic molecules on graphite surfaces provides another demonstration of the MARTINI force field limitations, as only the semi-quantitative agreement with the experimental data has been reached [167]. Although the entropy-enthalpy imbalance and transferability are common problems among different CG models, the universal nature of the MARTINI force field makes it not the most optimal choice for specific applications requiring finer detail.

### 4.2 Bottom-up methods

In contrast to the top-down approach, where the CG interactions are directly parameterized to fit experimental data, the bottom-up methods, also known as systematic coarse-graining, fit the CG interactions to reproduce the microscopic quantities obtained from atomistic simulations [4; 168–172]. Atomistic models are more accurate and have more chemical specificity, which means that having a CG potential reproducing a behavior of such an atomistic model can be of great value. Several bottom-up methods have been proposed over the years, most notably the force-matching [173-175], relative entropy minimization [5], iterative Boltzmann inversion (IBI) [176-177] and inverse Monte Carlo (IMC).
The IMC method, which is a structure-based coarse-graining method fitting the CG potential to reproduce the radial distribution functions (RDFs) obtained in an atomistic simulation has been used to simulate complex biological and macromolecular systems such as lipid bilayers and vesicles [160], DNA-protein complexes [180] and nucleosome core particles [181] [182]. This method, like the other bottom-up approaches, does not require any experimental data and works exclusively with the data obtained with a higher-resolution simulation. It is not limited by a specific coarse-grained mapping, like the MARTINI force field, for example, and the coarse-graining can be performed in successive steps, reaching “super” coarse-grained level with extremely low resolution but preserving the original detail [182]. Moreover, it has been shown that all studied IMC-derived CG potentials of phospholipids, obtained for unordered water-lipid systems with different compositions, could reproduce the structure of a lipid bilayer with reasonable values of the average area per lipid, orientational order, and compressibility, demonstrating a certain degree of CG potential transferability across different compositions, even in implicit solvent [160]. The theory behind the IMC method can be found in detail in these works [168] [180] [183]–[185], but a short description is also outlined below.

The formal definition of a CG potential $U^{CG}$ reproducing the structural properties of an all-atom (AA) system can be obtained in the following steps. First, we define a mapping operator $f(r^N) = R^M$ that transforms $N$ atomic positions into $M$ CG sites. An example of such mapping could be a computation of the center of mass for each subset of atomic positions $r^N$, where each center of mass defines the position of a CG site. The probability density functions (Equation 3.8) for both AA and CG systems are related to the corresponding potential functions:

$$\rho^{AA}(r^N) \propto \exp \left[ -\beta U^{AA}(r^N) \right]$$

$$\rho^{CG}(R^M) \propto \exp \left[ -\beta U^{CG}(R^M) \right]$$

(4.1)

We can reformulate the CG probability density function in terms of atomic configurations that map to the coarse-grained configuration by integrating over the atomic degrees of freedom:

$$\rho^{CG}(R^M) = \int d r^N \rho^{AA}(r^N) \delta [f(r^N) - R^M]$$

(4.2)

where the delta function ensures that only those atomic configurations that map to the coarse-grained configuration are included. By equating
the atomistic and CG probability density functions in Equation 4.1, we obtain the definition of the CG potential $U^{CG}$ in terms of the AA potential $U^{AA}$, which results in an N-body potential of mean force (PMF), equal to the free energy of the removed degrees of freedom:

$$U^{CG}(R^M) = -\frac{1}{\beta} \ln \int dr^N \exp \left[ -\beta U^{AA}(r^N) \right] \delta \left[ f(r^N) - R^M \right] + C \quad (4.3)$$

Note that the CG potential depends on the temperature and composition of the reference higher-resolution system. Running a CG simulation with such an N-body potential is not practical, however, and one usually approximates it with a set of pair potentials:

$$U^{CG}(R^M) \approx \sum_{i<j} U^{eff}_{ij}(R_{ij}) \quad (4.4)$$

An addition of angle and dihedral terms to Equation 4.4 is often used to improve the behavior of long molecular chains. Now the task is to find the best possible fit of the CG effective potentials (Equation 4.4), defined by a set of $L$ potential parameters $\{\lambda_\gamma\}$ ($\gamma = 1, 2, ..., L$), that reproduces a set of $L$ statistical averages $\{\langle A^{ref}_\alpha(r^N) \rangle\}$ ($\alpha = 1, 2, ..., L$) of a target property $A(r^N)$. The reference properties $\{\langle A^{ref}_\alpha(r^N) \rangle\}$ are usually recomputed at the CG resolution by mapping the atomistic coordinates $r^N$ in the reference simulation to the CG ones ($R^M$). Knowing the set of potential parameters $\{\lambda_\gamma\}$ allows us to compute the statistical averages $\{\langle A_\alpha(R^M) \rangle\}$ in a coarse-grained molecular dynamics or Metropolis Monte Carlo simulations. But finding the potential parameters $\{\lambda_\gamma\}$ from some known statistical averages is a more challenging task. In this context, it is often named as the inverse problem. To solve it, we start by relating the small changes of the potential parameters and the subsequent changes of the statistical averages $\langle A_\alpha(R^M) \rangle$.

$$\Delta \langle A_\alpha \rangle = \sum_\gamma \frac{\partial \langle A_\alpha \rangle}{\partial \lambda_\gamma} \Delta \lambda_\gamma + O(\Delta \lambda^2_\gamma) \quad (4.5)$$

The matrix element $\frac{\partial \langle A_\alpha \rangle}{\partial \lambda_\gamma}$ within the Jacobian matrix $J$ can be obtained by using the statistical mechanical expression for the ensemble average (Equation 3.10). If the property $A$ depends only on the positions ($R^M$) then:
\[
\frac{\partial \langle A_\alpha \rangle}{\partial \lambda_\gamma} = \frac{\partial}{\partial \lambda_\gamma} \frac{\int dR^M A_\alpha(R^M) \exp \left[ -\beta U(R^M) \right]}{\int dR^M \exp \left[ -\beta U(R^M) \right]} = -\beta \left( \langle A_\alpha \frac{\partial U}{\partial \lambda_\gamma} \rangle - \langle A_\alpha \rangle \langle \frac{\partial U}{\partial \lambda_\gamma} \rangle \right)
\]

(4.6)

where the resulting terms can be estimated in a CG simulation. Starting from a trial set of potential parameters \( \{\lambda_\gamma^{(0)}\} \), one can run a CG simulation to determine statistical averages of the corresponding properties \( \{\langle A_\alpha^{(0)} \rangle\} \) and compute the difference of the computed averages with the reference values \( \{\langle A_\alpha^{\text{ref}} \rangle\} \):

\[
\Delta \langle A_\alpha \rangle = \langle A_\alpha^{\text{ref}} \rangle - \langle A_\alpha^{(0)} \rangle
\]

(4.7)

By inverting the Jacobian and ignoring higher-order terms in the series expansion in Equation 4.5 one can compute the necessary corrections to the potential parameters:

\[
\Delta \lambda = \mathbf{J}^{-1} \Delta \langle A \rangle
\]

(4.8)

where \( \Delta \lambda \) and \( \Delta \langle A \rangle \) are vectors containing potential corrections and differences in the statistical averages at all points \( \gamma \) and \( \alpha \) and \( \mathbf{J}^{-1} \) is the inverted Jacobian matrix. The computed potential corrections are then applied to the trial set of potential parameters to obtain a better guess for the CG potential parameters:

\[
\lambda_\gamma^{(1)} = \lambda_\gamma^{(0)} + \eta \Delta \lambda_\gamma
\]

(4.9)

where \( \eta \) is the regularization parameter set from 0 to 1. The process of computing potential corrections can be performed iteratively until the difference between the computed and reference averages is minimized and convergence is reached. At the beginning of such optimization, the value of \( \eta \) is usually set to a lower value which improves the convergence. After reaching the convergence in \( n \) iterations, one obtains a set of potential parameters \( \{\lambda_\gamma^{(n)}\} \) that reproduces a configurational property \( A \) obtained in a reference atomistic simulation by approximating the N-body PMF shown in Equation 4.3. However, this approach can be applied outside of the CG potential parameterization and can be used to approximate classical atomistic potentials from \textit{ab initio} data or to optimize force field parameters [186]. In most cases where the IMC method is used,
one approximates the potential energy of a CG system by a tabulated potential of this form:

$$U^{CG} = \sum_{\alpha} S_{\alpha} \lambda_{\alpha}$$  \hspace{1cm} (4.10)

where $S_{\alpha}$ is a pair distance histogram element $\alpha$. The distance histogram is a configurational property and is connected to the radial distribution function (RDF):

$$g(r) = \frac{S(r)}{2\pi r^2 \Delta r \frac{V}{N(N-1)}}$$  \hspace{1cm} (4.11)

where $r$ is the pair distance, $\Delta r$ is the width of a spherical layer, $V$ is the volume and $N$ is the number of particles, $S(r)$ is the full distance histogram, which shows how many particle pairs are found with distances from $r$ to $r + \Delta r$. Using such formulation for the CG potential (Equation 4.10) one obtains the following expression of the matrix element shown in Equation 4.6:

$$\frac{\partial \langle S_{\alpha} \rangle}{\partial \lambda_{\gamma}} = -\beta \left( \langle S_{\alpha} S_{\gamma} \rangle - \langle S_{\alpha} \rangle \langle S_{\gamma} \rangle \right)$$  \hspace{1cm} (4.12)

Similar expressions are obtained if the distance histogram is replaced with a bond distance or angle distribution function. Metropolis Monte Carlo is the simulation method of choice within the IMC method, as it is stable to discontinuities in the potential and easy to parallelize efficiently. Moreover, using an implicit solvent reduces the number density of the CG system drastically, which allows large displacements in MC steps, improving the sampling efficiency. One of the important advantages of the IMC method is that it accounts for cross-correlations between pair interactions through the computation of the full Jacobian matrix in Equation 4.12. The IMC method is implemented in the MagiC software package [184] that incorporates a parallel MC simulation engine, the inverse problem solver, and tools for CG mapping and RDF computation. The resulting CG potentials can be imported into various MD simulation software packages such as GROMACS [187] and LAMMPS [142] to run CG MD simulations.

### 4.3 Neural network coarse-graining

Similarly to the neural network force fields or neural network potentials (NNPs) discussed in the previous chapter, data-driven approaches
can be employed to model coarse-grained systems within the bottom-up strategy, reproducing the behavior of atomistic systems. However, an important difference between the NNPs trained with quantum chemical data to produce a neural atomistic model and the neural network based coarse-grained models is that instead of learning the \textit{ab initio} potential energy surface, the neural CG models have to learn the free energy surface of the lost degrees of freedom or the N-body PMF, which is not available from the higher-resolution simulations directly \cite{172}. That means that it is not possible to train a CG NNP based on atomistic energies or forces only – instead, the neural networks learn the N-body PMF or its gradient (the forces). One notable exception is for the CG NNPs that reproduce a predefined coarse-grained potential and thus can learn on energies and even use a Behler-Parrinello approach \cite{183,189}. Some early attempts in designing the bottom-up CG NNP included learning the free energy surface directly, albeit of relatively low-dimensional systems of small peptides \cite{190,191}. Another possibility is to use the force-matching method, where the NN learns the gradient of the N-body PMF \cite{192,197}. Wang et al. \cite{193} have developed the CGnets deep learning approach that was used to simulate a small protein in water at CG resolution, without any explicit solvent. A further study has shown that many-body effects play a significant role for such a protein molecule at CG resolution, where up to 5-body terms are necessary to reproduce the atomistic behavior \cite{196}. The DeepCG framework was used to simulate a CG water with high accuracy \cite{192}. Apart from the force-matching, the relative entropy minimization method was used to train a CG NNP in a recent study by Thaler et al. \cite{198}. The Gaussian Approximation Potentials have been similarly employed to develop various CG models of bulk liquids and small biomolecules \cite{185,199,200}. Advanced NN architectures, such as auto-encoders \cite{194}, graph neural networks \cite{195,197,201}, and generative adversarial networks (GANs) \cite{202} have been used to predict CG potentials, implicit solvent potentials \cite{197} and even construct efficient mappings to both coarse-grained and atomistic resolutions \cite{194,203,205}. Moreover, a popular technique used in active learning called the query by committee \cite{206} has been used to improve the reliability of the potentials outside of the equilibrium configuration range \cite{126,172,207}. More in-depth information can be found in the recent reviews \cite{171,172,208,210}. 

\section*{50}
5. Summary of the results

The following chapter summarizes the results presented in Papers I-IV, included in this thesis. The primary focus of Paper I is elucidating the mechanisms of chronic pulmonary inflammations caused by nanomaterial exposure, observed in toxicological in vivo and in vitro studies. This work is an outcome of multidisciplinary research that combines advanced imaging, experimental, and simulation techniques, performed by more than 30 authors. The contribution of our research group consisted of molecular dynamics simulations of different TiO$_2$ surfaces with cell membrane lipid molecules that confirmed the strong affinity of the surfaces towards the lipids and demonstrated how the layered nanomaterial-lipid structures can form. In Paper II, we continued studying the lipid adsorption on titania surfaces in more detail, revealing various lipid binding modes and their prevalence based on the type of the surface and lipid headgroup. Paper III describes a bottom-up coarse-grained model of TiO$_2$ surfaces interacting with phospholipids that is built on the data from the previous study. Finally, in Paper IV we explore an alternative approach to bottom-up coarse-graining using the neural network potentials trained on structural data in an attempt to improve the model transferability.

5.1 Atomistic simulation studies of phospholipid adsorption on TiO$_2$ surfaces (I, II)

This section focuses on the atomistic molecular dynamics simulations of phospholipids near different TiO$_2$ surfaces, presented in detail in Paper II and as our contribution to a larger work published in Paper I. The latter studied how lung epithelial cells react to nanomaterial exposure. It has been found that soon after the exposure to TiO$_2$ NPs, the nanomaterial is internalized and relocated to the cell surface in the form of NP-lipid aggregates, called “cauliflowers” because of their shapes in the recorded fluorescence micrographs. This process of nanomaterial concentration in the form of cauliflowers was given a name “nanoquaran-
Nanoquarantining is accompanied by increased lipid synthesis that facilitates cell survival after exposure to high doses of NPs, where the ratio of nanomaterial surface to the cell surface can go up to 100:1. It has been shown that the nanoquarantining is a consequence of active cellular response. Blocking the fatty acid synthase enzyme, which takes part in lipid synthesis, by drug resveratrol inhibits the formation of large cauliflowers. Thus, nanoquarantining can be viewed as a defense mechanism of the epithelial cells. The formation of cauliflowers triggers the immune response by macrophages that attack epithelial cells containing the nanomaterial. However, the macrophages, lacking the ability to process the NPs to cauliflowers themselves, die from the exposure if a large amount of the nanomaterial has accumulated. The newly released nanomaterial is taken again by the epithelial cells, thus closing the cycle of chronic inflammation. The whole nanomaterial cycle is outlined in Figure 5.1.

Figure 5.1: Scheme of the discovered cycle of nanomaterial in the lung alveolar surface model with associated key molecular events that drive chronic inflammation following the exposure to nanomaterials together with a legend of graphical elements (pictograms). Individual events are described in more detail in Figures 2-5 of the original paper. Provided by experimental collaborators. Reprinted from Paper I.

Evidently, the role of nanomaterial-lipid interactions is crucial in the phenomenon of nanomaterial cycling. Figure 2.2 containing the TEM images of TiO$_2$ NPs with phospholipid liposomes shown in Chapter 2 demonstrates the overall affinity of TiO$_2$ NPs to phospholipid molecules, but molecular simulations can resolve the exact mechanisms of the lipid
adsorption on TiO$_2$ surfaces. For example, Figure 5.2 shows how initially unordered lipid molecules near an anatase (101) surface, one of the most prevalent lattice planes in anatase NPs [211], form a lipid bilayer adsorbed to the TiO$_2$ surface. This illustrates that the lipids attach to the surface through the polar headgroups, which enables more nanosurfaces to attach to the opposite side of the lipid bilayer, suggesting an explanation for the layered nature of the nanomaterial-lipid cauliflowers.

![Figure 5.2: Formation of a lipid bilayer on TiO$_2$ surface during a 1 µs long MD simulation. Water and ions are not shown. Adapted from Paper II.](image)

Moreover, the MD simulations show that different headgroups adsorb to the surface through different binding modes as shown in Figure 5.3. For example, the POPE lipid, having a protonated amino group can bind to the surface through hydrogen bonding at a close separation. However, the DMPC lipid has a choline group that cannot form hydrogen bonds. Instead, the lipid attaches to the surface through a water-mediated contact of a phosphate group with the surface. Different binding modes have different relative adsorption strengths, which has direct relevance for the structure and stability of the cauliflowers.

![Figure 5.3: POPE (left) and DMPC (right) lipids adsorbed on anatase (101) surface. Water and ions are not shown. Obtained from an MD simulation. Adapted from Paper I.](image)

Thus, in continuation to the molecular dynamics simulations of TiO$_2$ surfaces with lipids shown in Paper I, we perform a more elaborate study of the adsorption of different types of lipids to several low-energy TiO$_2$
surfaces in Paper II. The studied lipids are DMPC, POPE, and cholesterol (CHL) shown in Figure 5.4, and TiO$_2$ surfaces are rutile (101), rutile (110), anatase (100), anatase (101) and a small spherical anatase nanoparticle with radius of 2 nm, all shown in Figure 5.5. The detailed composition of all simulated systems is shown in Table 5.1.

<table>
<thead>
<tr>
<th>DMPC</th>
<th>POPE</th>
<th>CHL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glycerol moiety</td>
<td>Glycerol moiety</td>
<td>Hydroxyl group</td>
</tr>
<tr>
<td>Phosphate</td>
<td>Phosphate</td>
<td>Ethanolamine</td>
</tr>
<tr>
<td>Choline</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 5.4:** Lipid molecules used in MD simulations with TiO$_2$ surfaces. Adapted from Paper II.

<table>
<thead>
<tr>
<th>Rutile (101)</th>
<th>Rutile (110)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anatase (100)</td>
<td>Anatase (101)</td>
</tr>
<tr>
<td>Anatase NP</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 5.5:** TiO$_2$ surfaces used in MD simulations with lipids. Adapted from Paper II.

To describe the interactions, we have used the Slipids force field [101, 212] for lipids, an *ab initio* derived force field for TiO$_2$ [213], TIP3P water model [214] and Yoo and Aksimentiev parameters for sodium and chloride ions [215]. Hydrated TiO$_2$ surfaces at neutral pH are covered with hydroxyl groups and are negatively charged [216, 218]. Furthermore, *ab initio* MD simulations show that TiO$_2$ can split water molecules
Table 5.1: Composition of the simulated systems at atomistic resolution.
Adapted from Paper II.

<table>
<thead>
<tr>
<th>System</th>
<th>Lipids</th>
<th>N(H₂O)</th>
<th>Ions</th>
<th>Box size, nm</th>
<th>t, µs</th>
</tr>
</thead>
<tbody>
<tr>
<td>anatase (101)</td>
<td>82 POPE</td>
<td>11785</td>
<td>94 Na⁺+33 Cl⁻</td>
<td>7.1x6.9x12.4</td>
<td>6</td>
</tr>
<tr>
<td>anatase (101)</td>
<td>82 DMPC</td>
<td>12042</td>
<td>94 Na⁺+33 Cl⁻</td>
<td>7.1x6.9x12.4</td>
<td>1</td>
</tr>
<tr>
<td>anatase (100)</td>
<td>76 POPE</td>
<td>11186</td>
<td>92 Na⁺+31 Cl⁻</td>
<td>7.1x6.4x12.6</td>
<td>1</td>
</tr>
<tr>
<td>anatase (100)</td>
<td>76 DMPC</td>
<td>11313</td>
<td>92 Na⁺+31 Cl⁻</td>
<td>7.1x6.4x12.5</td>
<td>1</td>
</tr>
<tr>
<td>rutile (110)</td>
<td>85 POPE</td>
<td>12025</td>
<td>97 Na⁺+33 Cl⁻</td>
<td>7.1x7.3x12.2</td>
<td>1</td>
</tr>
<tr>
<td>rutile (110)</td>
<td>85 DMPC</td>
<td>12245</td>
<td>98 Na⁺+34 Cl⁻</td>
<td>7.1x7.3x12.2</td>
<td>1</td>
</tr>
<tr>
<td>rutile (101)</td>
<td>82 POPE</td>
<td>11414</td>
<td>127 Na⁺+32 Cl⁻</td>
<td>7.0x7.2x11.9</td>
<td>1</td>
</tr>
<tr>
<td>rutile (101)</td>
<td>82 DMPC</td>
<td>11642</td>
<td>127 Na⁺+32 Cl⁻</td>
<td>7.0x7.2x11.8</td>
<td>1</td>
</tr>
<tr>
<td>anatase (101)</td>
<td>120 POPE</td>
<td>11800</td>
<td>94 Na⁺+33 Cl⁻</td>
<td>7.1x6.9x13.4</td>
<td>1</td>
</tr>
<tr>
<td>anatase (101)</td>
<td>120 DMPC</td>
<td>12145</td>
<td>95 Na⁺+34 Cl⁻</td>
<td>7.1x6.9x13.3</td>
<td>1</td>
</tr>
<tr>
<td>anatase NP</td>
<td>83 POPE</td>
<td>93391</td>
<td>309 Na⁺+259 Cl⁻</td>
<td>14.4x14.4x14.4</td>
<td>1</td>
</tr>
<tr>
<td>anatase (101)</td>
<td>82 POPE+16 CHL</td>
<td>11472</td>
<td>93 Na⁺+32 Cl⁻</td>
<td>7.1x6.9x12.5</td>
<td>1</td>
</tr>
<tr>
<td>anatase (101)</td>
<td>82 DMPC+16 CHL</td>
<td>11647</td>
<td>93 Na⁺+32 Cl⁻</td>
<td>7.1x6.9x12.4</td>
<td>1</td>
</tr>
</tbody>
</table>

Thus, in our simulations, we attach hydroxyl groups to 30% of five-coordinated Ti atoms and all four-coordinated Ti atoms. The surface modification results in a slight negative charge that is within the limits of the reported experimental data. The TiO₂ slabs are periodic in the lateral directions, allowing the lipids to attach only to the “top” or the “bottom” sides of the slab with the specific lattice planes, e.g. (110), (101). Even though no position restraints were applied, the slabs remained fixed during the simulations, with the atoms only fluctuating around their positions. However, the anatase nanoparticle freely moved as a whole during the simulation. The boxes are equilibrated at 303 K and 1 bar, but the production simulations are run at NVT ensemble (with fixed volume). Most of the runs continue for 1 µs (see Table 5.1). The MD simulations are performed using the GROMACS 2019 simulation package [187] and the visualizations are done using VMD [221].

To study the adsorption of lipids, we have calculated the number density profiles of specific lipid headgroup atoms with respect to the distance to the surface. The number density profiles are computed by averaging the atom-surface distance histograms over all lipid molecules and frames and normalizing them to the volume of a box slice. In addition, to track the specific binding modes, we have computed the number density of the closest lipid headgroup atom to the surface. An example of such number density profiles is shown in Figure 5.6. We identify the binding modes by looking at the number density profile of the closest atom and comparing it with the atomic number density profiles. For example, the peak A of Figure 5.6 coincides with the number density peak of the O14 atom of the phosphate group at 2.2 Å, which means that at this distance, only
the O14 atom approaches the surface, thus contributing significantly to the lipid adsorption. It is a comparatively rare binding event, however, as the corresponding peak area is small. The explanation for the low frequency of such a binding mode is that it involves direct contact of the phosphate group with the surface that follows unfavorable desorption of the surface water layer [219, 220]. However, the next closest atom number density peak (B) coincides with a large peak of nitrogen atom density that comes from the adsorbed amino group, which is held by hydrogen bonds with the surface [222]. Finally, the last peak of the closest atom number density (C) coincides again with the phosphate group oxygen atoms, now with both O14 and O13, and corresponds to the water-mediated phosphate group contact. Further analysis of the number density profiles allows us to compute the fraction of adsorbed lipids and their decomposition to various binding modes. Such a fraction can also be interpreted as a probability of a lipid to be adsorbed through a specific binding mode, or $P_b$, and can be computed as follows:

$$
P_b = \frac{\int_{r_1}^{r_2} n(r) dr}{\int_0^{r_{\text{max}}} n(r) dr}
$$

(5.1)

where $n(r)$ is the number density of the closest atom, $r_1$ and $r_2$ are the number density peak boundaries and $r_{\text{max}}$ is the maximum distance from the TiO$_2$ surface (in our case $r_{\text{max}} = 4.5$ nm). In addition, we can estimate the lower bound of the mean residence time of an adsorbed lipid on the surface for each binding mode by measuring how long the lipids stay adsorbed in a certain binding mode. An example of the residence time histograms for the ethanolamine and water-mediated phosphate binding modes of POPE lipid on the anatase (101) surface is shown in Figure 5.7.

When examining the TiO$_2$-DMPC systems we have additionally found the glycerol moiety and the choline group binding modes, but no specific binding modes of cholesterol have been found, as in our simulations it stayed in the tail region of the lipid aggregates, avoiding the surface contact. Table 5.2 and 5.3 show the calculated probabilities of each binding mode $P_b$ together with the mean residence times. Note that the $P_b$ is computed as a percentage of the total number of phospholipids (cholesterol is excluded). We neither present the results on the direct phosphate contact, nor the glycerol moiety binding due to an insufficient statistical sample compared with the other binding modes. In the case of POPE lipid adsorption, the ethanolamine binding mode is particularly strong on the anatase (100) and rutile (110) surfaces, both in terms of the frac-
tion of lipids adsorbed through this mode and the mean residence time. The water-mediated phosphate binding mode is very common too, but with a shorter residence time in most cases. It is present in the DMPC adsorption as well but with both shorter residence time and a smaller number density peak area. Choline binding is also present in DMPC, but has an even shorter residence time, except for direct contact on the rutile (101) surface. High curvature of the small anatase NP likely prevents strong lipid adsorption, as the lipid aggregate has to bend to cover the curved surface of the NP, resulting in significantly reduced fractions of adsorbed lipids and residence times. Cholesterol has a different effect on adsorption, simultaneously increasing the binding through ethanolamine in POPE and choline in DMPC while reducing the binding through the phosphate group, as cholesterol decreases the area per lipid in bilayers [223; 224], favoring the adsorption through an end group, such as choline or ethanolamine. Finally, by integrating the closest atom number density profiles in all systems up to 1 nm, which roughly corresponds to a linear size of the headgroup in a lipid bilayer, we estimate the total fractions of adsorbed lipids and rank the TiO$_2$ surfaces by their affinities towards DMPC and POPE lipids. The calculated fractions are shown in Table 5.4. The data suggests that the POPE adsorption is considerably stronger than the adsorption of DMPC lipids. However, each surface has a different relative preference for each lipid. Based on our calculations of the total fractions of adsorbed lipids, we estimate that the adsorption strength of POPE lipids increases in the sequence anatase (NP) << rutile (101) ≈ rutile (110) < anatase (101) < anatase (100). Similarly, for DMPC lipids – anatase (100) << rutile (110) < anatase (101) < rutile (101) with the order essentially reversed.

In summary, the atomistic molecular dynamics simulations of cell membrane lipids near different TiO$_2$ surfaces have supported experimental investigations of living cell cultures exposed to nanomaterials presented in Paper I. In particular, our simulations have demonstrated that the lipids binding to TiO$_2$ surfaces in the form of bilayers through the headgroups enable the formation of layered nanomaterial-lipid structures observed experimentally. Then, we continue our work in Paper II, revealing different possibilities for the lipid molecules to bind to various TiO$_2$ surfaces and performing the estimates of the lipid adsorption strength. We find that the type of lipid, a type of TiO$_2$ crystal polymorph and the lattice plane have profound effects on the overall adsorption. The data generated in the atomistic molecular dynamics simulations is used to develop a bottom-up coarse-grained model of lipids near TiO$_2$ surfaces, as shown in Paper III.
Figure 5.6: Anatase (101) slab-POPE headgroup atoms number density profiles and corresponding representations of the binding modes. A: Phosphate group (direct contact); B: Ethanolamine group; C: Phosphate group (water-mediated). Other lipid molecules and surrounding water and ions are removed from the background of the molecular images for clarity. Adapted from Paper II.
Figure 5.7: Histograms of residence times for two POPE binding modes on anatase (101) surface. The y-axis shows the natural logarithm of the occurrence.

Table 5.2: POPE binding modes characteristics. Adapted from Paper II.

<table>
<thead>
<tr>
<th>Binding mode</th>
<th>System</th>
<th>$P_b$ %</th>
<th>Residence time, ns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethanolamine</td>
<td>rutile (110)</td>
<td>28.5</td>
<td>19.9</td>
</tr>
<tr>
<td></td>
<td>rutile (101)</td>
<td>12.1</td>
<td>2.1</td>
</tr>
<tr>
<td></td>
<td>anatase (100)</td>
<td>35.1</td>
<td>36.6</td>
</tr>
<tr>
<td></td>
<td>anatase (101)</td>
<td>18.2</td>
<td>8.1</td>
</tr>
<tr>
<td></td>
<td>anatase NP</td>
<td>7.2</td>
<td>4.1</td>
</tr>
<tr>
<td></td>
<td>anatase (101) + CHL</td>
<td>21.4</td>
<td>10.1</td>
</tr>
<tr>
<td>Phosphate (water-mediated)</td>
<td>rutile (101)</td>
<td>23.3</td>
<td>3.1</td>
</tr>
<tr>
<td></td>
<td>anatase (100)</td>
<td>13.3</td>
<td>7.7</td>
</tr>
<tr>
<td></td>
<td>anatase (101)</td>
<td>21.0</td>
<td>6.0</td>
</tr>
<tr>
<td></td>
<td>anatase NP</td>
<td>5.8</td>
<td>3.2</td>
</tr>
<tr>
<td></td>
<td>anatase (101) + CHL</td>
<td>18.7</td>
<td>5.5</td>
</tr>
</tbody>
</table>
Table 5.3: DMPC binding modes characteristics. Adapted from Paper II.

<table>
<thead>
<tr>
<th>Binding mode</th>
<th>System</th>
<th>$P_b, %$</th>
<th>Residence time, ns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Choline (direct)</td>
<td>rutile (101)</td>
<td>15.7</td>
<td>3.5</td>
</tr>
<tr>
<td>Phosphate (water-mediated)</td>
<td>rutile (110)</td>
<td>8.5</td>
<td>3.9</td>
</tr>
<tr>
<td></td>
<td>rutile (101)</td>
<td>5.2</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>anatase (100)</td>
<td>1.6</td>
<td>1.8</td>
</tr>
<tr>
<td></td>
<td>anatase (101)</td>
<td>13.4</td>
<td>5.8</td>
</tr>
<tr>
<td></td>
<td>anatase (101) + CHL</td>
<td>10.7</td>
<td>3.9</td>
</tr>
<tr>
<td>Choline (water-mediated)</td>
<td>rutile (110)</td>
<td>10.7</td>
<td>1.3</td>
</tr>
<tr>
<td></td>
<td>rutile (101)</td>
<td>8.2</td>
<td>1.2</td>
</tr>
<tr>
<td></td>
<td>anatase (100)</td>
<td>2.4</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td>anatase (101)</td>
<td>7.9</td>
<td>1.3</td>
</tr>
<tr>
<td></td>
<td>anatase (101) + CHL</td>
<td>9.3</td>
<td>1.3</td>
</tr>
</tbody>
</table>

Table 5.4: Total fraction of adsorbed lipids for TiO$_2$ nanosurfaces at atomistic resolution. The lipid is considered adsorbed if the closest headgroup atom is within 1 nm from the surface. Adapted from Paper II.

<table>
<thead>
<tr>
<th>TiO$_2$ surface</th>
<th>% of adsorbed POPE</th>
<th>% of adsorbed DMPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>rutile (110)</td>
<td>42.9</td>
<td>22.2</td>
</tr>
<tr>
<td>rutile (101)</td>
<td>42.9</td>
<td>34.3</td>
</tr>
<tr>
<td>anatase (100)</td>
<td>53.0</td>
<td>5.0</td>
</tr>
<tr>
<td>anatase (101)</td>
<td>46.5</td>
<td>28.4</td>
</tr>
<tr>
<td>anatase NP</td>
<td>22.8</td>
<td>-</td>
</tr>
</tbody>
</table>
5.2 Development of a bottom-up coarse-grained model for phospholipids and TiO$_2$ surfaces (III)

Paper III describes a methodology behind a bottom-up coarse-grained model, obtained from the atomistic molecular dynamics simulations of lipids near TiO$_2$ surfaces presented in Paper I and II using the inverse Monte Carlo method, described in Chapter 4 of this thesis. The CG model enables significant computational resource savings while capturing important structural details of the atomistic simulations without using any experimental data. The model is tested on a larger scale using coarse-grained molecular dynamics simulations (CGMD), which is particularly relevant to the nanomaterial-lipid interactions, as evident from the nanotoxicity overview given in Chapter 2 as well as Paper I. Specifically, it mentions a significant difference in the toxic effect of anatase nanoparticles of different shapes – long rods (cylinders) and nanocubes. While atomistic simulations can describe the nanomaterial-lipid interactions for periodic surface slabs and small nanoparticles, the effect of nanomaterial shape at a scale of tens of nanometers is significantly harder and more expensive to resolve at atomistic resolution. By systematically reducing the number of degrees of freedom, we achieve similar simulation rates for coarse-grained systems that are over fifty times as large by volume when compared to atomistic small-scale simulations. However, the resulting speed-up is even greater due to the significantly reduced friction at the coarse-grained resolution, as discussed in Chapter 4.

The first step in developing a bottom-up CG model is to map the atomistic coordinates to the coarse-grained ones, in a process known as bead mapping. An illustration of such bead mapping is shown in Figure 5.8. For lipids, we use a similar bead mapping reported in earlier works from our research group [160] that is also reminiscent of MARTINI bead mapping for lipids. For TiO$_2$ slabs, we devise a special bead mapping algorithm that groups the atoms based on their positions in a regular grid separating the TiO$_2$ into multiple subunits of 5-8 atoms each. The bulk of TiO$_2$ is excluded, leaving only two layers of CG beads on each side of the slab. The beads in the outermost layer (TiA type) are assigned a uniform negative charge that reproduces TiO$_2$ surface charge density, and the subsurface layer (TiB type) serves as a barrier for lipid molecules and ions, preventing them from reaching the bulk of the nanomaterial. The ions remain unchanged at the CG resolution, but no water beads are used – the solvent is described implicitly in our CG models.

The next step is to find the effective CG potential that would approxi-
mate the N-body PMF (Equation 4.3) and reproduce the structural properties – the radial distribution functions (RDF) together with the bond and angle distribution functions from the small-scale reference atomistic simulations. We use the IMC method, where the non-electrostatic part of the effective potential is optimized iteratively during several iterations. An example of an iterative optimization for a pair surface TiO$_2$ site (TiA) – ethanolamine bead (NEL) is shown in Figure 5.9. During each IMC iteration, the current CG potential is tested by running a Monte Carlo simulation, sampling the RDF, and comparing it to the reference data from the atomistic simulations. The corresponding potential correction is applied before moving to the next iteration. In this particular example given in Figure 5.9, the RDF agreement steadily improves until almost no difference between the reference RDF and the RDF reproduced by the CG potential is noticeable. The resulting CG potentials are saved in a tabular form which is later imported into the LAMMPS format to run the CGMD simulations. The forces on each CG bead are obtained by computing the derivatives of the tabulated potential with linear interpolation between the potential points. The IMC procedure is performed for all TiO$_2$ slab-lipid systems reported in Paper II, except for the anatase (100)-DMPC system due to weak adsorption, and
the systems with cholesterol. After the CG potentials are optimized, we run the CGMD simulations of the systems at the original scale to validate the potentials by comparing the CGMD results to the all-atom MD simulations, before moving on to the larger scale. To further speed up the CGMD simulations, we substitute the tabulated bond and angle potentials with their harmonic approximation and compare the results obtained for the original IMC-derived bond and angle potentials and their harmonic fit. The IMC procedure, the bead mapping, and the analysis are performed with the MagiC v.3 software package [184] and the CGMD simulations are carried out using LAMMPS [142] (29 Oct 2020 version). The initial configurations of the large-scale systems are prepared using Packmol [225] and the molecular images are generated with VMD [221].

**Figure 5.9:** Convergence of RDFs during the IMC procedure (left) and the corresponding CG potentials (right) for rutile (110) with POPE. Adapted from Paper III.

After the RDF differences given by the optimized CG potentials and the reference data are minimized, we validate the CG potentials by running the CGMD simulations and comparing the resulting number density profiles of the lipid headgroup beads to the reference data mapped to the CG resolution, so that the comparison is done at the same scale. An example of such comparison is shown in Figure 5.10. The positions of the number density peaks are well reproduced by both CG models, except for the second peak of ethanolamine density. Surprisingly, the CG model with the harmonic fits (column C) for the bonded potentials gives a better agreement with the reference data (column A) than the pure IMC potentials (column B), reproducing the peak heights better as
well as giving the correct shape of the adsorbed lipid aggregate. Similar behavior is observed for the anatase (100)-POPE system, but no significant difference is found for other TiO$_2$-lipid systems, except for the rutile (110)-DMPC where the CG model with harmonic bonded potential does not reproduce a small peak of the phosphate density. The root-mean-square deviations of headgroup number density for all studied CG systems are shown in Table 5.5. Overall, our CG models show a good agreement with the number density profiles from the reference data, even though they were not used to optimize the CG potentials. Additionally, we compute the total fraction of adsorbed lipids using the method presented in Paper II for the atomistic simulations mapped to the CG resolution, CGMD simulations with pure IMC potentials, and the CGMD simulations with harmonic approximations for the bond and angle potentials. The results are shown in Table 5.6.

**Table 5.5:** Root-mean-square deviations of headgroup number density (nm$^{-3}$). Adapted from Paper III.

<table>
<thead>
<tr>
<th>System</th>
<th>Coarse-grained MD (Tabulated potentials)</th>
<th>Coarse-grained MD (Harmonic potentials)</th>
</tr>
</thead>
<tbody>
<tr>
<td>anatase (101)-POPE</td>
<td>0.2440</td>
<td>0.1326</td>
</tr>
<tr>
<td>anatase (101)-DMPC</td>
<td>0.0718</td>
<td>0.0743</td>
</tr>
<tr>
<td>anatase (100)-POPE</td>
<td>0.2482</td>
<td>0.1743</td>
</tr>
<tr>
<td>rutile (110)-POPE</td>
<td>0.1624</td>
<td>0.1348</td>
</tr>
<tr>
<td>rutile (110)-DMPC</td>
<td>0.0587</td>
<td>0.0930</td>
</tr>
<tr>
<td>rutile (101)-POPE</td>
<td>0.1320</td>
<td>0.1021</td>
</tr>
<tr>
<td>rutile (101)-DMPC</td>
<td>0.0748</td>
<td>0.0679</td>
</tr>
</tbody>
</table>

**Table 5.6:** Total fraction of adsorbed lipids for TiO$_2$ nanosurfaces at CG resolution. The lipid is considered adsorbed if the closest headgroup bead is within 1 nm from the surface. Adapted from Paper III.

<table>
<thead>
<tr>
<th>System</th>
<th>AAMD (Mapped to CG)</th>
<th>CGMD (Tabulated potentials)</th>
<th>CGMD (Harmonic potentials)</th>
</tr>
</thead>
<tbody>
<tr>
<td>anatase (101)-POPE</td>
<td>45.4 %</td>
<td>40.0 %</td>
<td>45.9 %</td>
</tr>
<tr>
<td>anatase (101)-DMPC</td>
<td>23.7 %</td>
<td>25.0 %</td>
<td>22.6 %</td>
</tr>
<tr>
<td>anatase (100)-POPE</td>
<td>52.5 %</td>
<td>49.8 %</td>
<td>53.9 %</td>
</tr>
<tr>
<td>rutile (110)-POPE</td>
<td>41.2 %</td>
<td>41.9 %</td>
<td>43.6 %</td>
</tr>
<tr>
<td>rutile (110)-DMPC</td>
<td>15.3 %</td>
<td>18.0 %</td>
<td>11.7 %</td>
</tr>
<tr>
<td>rutile (101)-POPE</td>
<td>41.6 %</td>
<td>43.6 %</td>
<td>41.1 %</td>
</tr>
<tr>
<td>rutile (101)-DMPC</td>
<td>29.9 %</td>
<td>30.1 %</td>
<td>29.2 %</td>
</tr>
<tr>
<td>anatase (101) (14 x 14 nm)-POPE</td>
<td>-</td>
<td>-</td>
<td>43.7 %</td>
</tr>
<tr>
<td>anatase NP (r = 2 nm)-POPE</td>
<td>21.5 %</td>
<td>-</td>
<td>25.6 %</td>
</tr>
<tr>
<td>anatase NP (r = 5 nm)-POPE</td>
<td>-</td>
<td>-</td>
<td>32.4 %</td>
</tr>
<tr>
<td>anatase NP (r = 10 nm)-POPE</td>
<td>-</td>
<td>-</td>
<td>39.7 %</td>
</tr>
</tbody>
</table>

*the simulation is not fully equilibrated, the given data is for the last 250 ns

After validating the CG potentials, we run the CGMD simulations
at a larger scale. Our large-scale systems consist of a larger anatase (101) periodic slab with an increased number of POPE molecules and ions, three systems with static spherical anatase NPs of different radii (2, 5, and 10 nm) surrounded by POPE lipids, as well as a POPE bilayer with a small mobile anatase NP (r = 2 nm). In the latter system, the NP is treated as a rigid body which allows it to move and rotate as a whole. All large-scale simulations contained 0.15 M NaCl in the water phase, or the volume that is free from lipids, ions, and the nanomaterial (including the bulk). The detailed composition of the large-scale CG systems is shown in Table 5.7. Our test simulations have shown that using the harmonic approximation for bonds and angles in the large-scale simulations provides a speed-up of 25-40 %, so in our production runs of the large-scale systems we only use the harmonic bonds and angles.

Table 5.7: Composition of large-scale CG systems. Adapted from Paper III.

<table>
<thead>
<tr>
<th>Anatase nanosurface</th>
<th>N(POPE)</th>
<th>Ions</th>
<th>Box size, nm</th>
<th>t, μs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slab (14 x 14 nm)</td>
<td>328</td>
<td>376 Na⁺+132 Cl⁻</td>
<td>14.0x14.0x13.0</td>
<td>1</td>
</tr>
<tr>
<td>NP (r = 2 nm)</td>
<td>83</td>
<td>309 Na⁺+259 Cl⁻</td>
<td>14.4x14.4x14.4</td>
<td>1</td>
</tr>
<tr>
<td>NP (r = 5 nm)</td>
<td>523</td>
<td>987 Na⁺+665 Cl⁻</td>
<td>20.5x20.5x20.5</td>
<td>1</td>
</tr>
<tr>
<td>NP (r = 10 nm)</td>
<td>2092</td>
<td>3642 Na⁺+2354 Cl⁻</td>
<td>32.0x32.0x32.0</td>
<td>1</td>
</tr>
<tr>
<td>NP⁺ (r = 2 nm)</td>
<td>882b</td>
<td>436 Na⁺+386 Cl⁻</td>
<td>16.5x16.5x20.0</td>
<td>1</td>
</tr>
</tbody>
</table>

a mobile NP
b in the form of a bilayer, pre-equilibrated for 100 ns

To build the large anatase slab, we distribute the TiO₂ CG beads (TiA and TiB) on four parallel planes, each with a surface area that is roughly four times as large as the original slab. In the larger slab, we preserve the average surface bead density, the spacings between TiA and TiB layers as well as the overall thickness. The system snapshots in Figure 5.11 illustrate a similar behavior of initially unordered lipids attaching to TiO₂ surface and forming a bilayer-like structure but at a bigger scale. Moreover, Table 5.6 shows that the total fraction of adsorbed lipids is very similar to the reference data at a smaller scale. The spherical NPs are built from two concentric spheres of TiA and TiB beads. To evenly distribute the beads on the spheres, we build Fibonacci lattices [226] of the beads and map them to the spheres. As with the large TiO₂ slab, we preserve the original surface density of the CG beads and the separation between TiA and TiB layers. To simulate the interactions of the spherical anatase NPs with POPE lipids, we use the
CG potentials obtained for the flat anatase (101)-POPE system with an increased charge on TiA beads that matches the higher surface charge density of the NPs. This can serve as a reasonable first approximation for the anatase NP interactions with lipids, as X-ray characterization of spherical anatase NPs shows that the anatase (101) plane is the most prevalent [211]. Furthermore, we can compare the results of CGMD simulations of a small anatase NP and POPE lipids with the atomistic MD simulation of the same system, performed in Paper II, mapped to the CG coordinates. The number density profiles and the snapshots are shown in Figure 5.12 and the corresponding total fractions of adsorbed lipids are shown in Table 5.6. Even though there are more intermediate number density peaks in the data from the atomistic simulation (row A), the positions of the main ethanolamine and phosphate peaks, and the shape of the adsorbed lipid aggregate are reproduced by the CGMD simulation with the potentials from flat anatase (101) slab with POPE (row B). However, the total fraction of adsorbed lipids is somewhat overestimated by the CG model. We note that the difference may arise from a rougher NP surface in the atomistic simulation and that a perfectly spherical shape of the NP at CG resolution is not an ideal representation of the surface, as it should contain separate faces with different lattice planes as well as edges and surface defects. This modification of the NP shape can be a matter of future studies. Figure 5.13 demonstrates how randomly oriented POPE lipids combine into a large curved bilayer, adsorbed to the surface of a 10 nm anatase NP. Table 5.6 reveals that the total fraction of adsorbed lipids on anatase NPs of different sizes increases with the radius of the NP. This is in line with the fact that the membrane bending, which increases together with the surface curvature, counteracts the adsorption [227]. It is thus natural to suggest, that in the absence of a strong lipid-surface defect interaction, the adsorption increases as the surface flattens. Figure 5.14 shows how a small mobile anatase NP first attaches to a POPE bilayer and later becomes partially wrapped by the bilayer, with a lipid protrusion appearing on the opposite side of the bilayer. Similar to the previous argument, such an outcome without the complete wrapping of the NP by the lipid bilayer is possible due to the high surface curvature of the small NP, which is consistent with the theoretical data [228].

Using the IMC bottom-up coarse-graining approach, we build an efficient CG model that reproduces important structural features of an atomistic system containing TiO_2 nanosurfaces and lipid molecules in an aqueous environment without any empirical data and explicit solvent. The resulting model can be enriched by a more realistic description of
the nanoparticles as well as an addition of more types of lipid molecules, most importantly cholesterol, as well as the cross-interaction terms to allow the simulation of complex lipid mixtures, present in real cell membranes. Such a model can be used to simulate the observed formation of nanomaterial-lipid aggregates inside the cells for different NP exposure scenarios involving multiple NPs of different shapes and realistic lipid mixtures. Moreover, the approach presented here can be extended to other types of nanomaterials and include even more complex molecules such as surfactants and polymers, but also more types of biomolecules – sugars, proteins, and DNA. However, we note the fairly limited transferability of each CG potential, obtained for a specific TiO$_2$-lipid pair. Even though it is possible to extend the model by adding more CG bead types, for example for different TiO$_2$ surfaces and other types of biomolecules, the model error would gradually increase as we go away from the original concentration of lipids, ions as well as the temperature. This issue of limited transferability is explored in Paper IV, presented in the next section, where we combine the ideas of the IMC method and the neural network potential to build a CG model of water-methanol solutions that is transferable over different concentrations.
Figure 5.10: Comparison of POPE lipid headgroup beads (NEL – ethanolamine, PCL – phosphate, COL – glycerol moiety) density profiles in the proximity of the TiO$_2$ surface (first row), closest bead density profile at a larger scale (second row) and the system snapshots (third row) of anatase (101)-POPE system. Column descriptions: (A) AA reference mapped to CG coordinates, (B) CG simulation with tabulated bonded potentials, (C) CG simulation with harmonic bonded potentials. Adapted from Paper III.
Figure 5.11: Large anatase slab (14 x 14 nm) with 328 POPE lipids during the CGMD simulation at: (A) t = 0 ns, (B) t = 1 ns, (C) t = 100 ns, (D) t = 1000 ns. Na\(^+\) and Cl\(^-\) ions are shown as yellow and light green spheres. Adapted from Paper III.
Figure 5.12: Comparison of POPE lipid headgroup number densities in the proximity of anatase spherical NP (r = 2 nm) surface (first column), closest lipid headgroup number density at a larger scale (second column) and the system snapshots (third column). Row descriptions: (A) atomistic reference mapped to CG coordinates, (B) CGMD with anatase (101)-POPE potential with harmonic fits. Adapted from Paper III.
Figure 5.13: Anatase spherical nanoparticle ($r = 10$ nm) with 2092 POPE lipids (ions are not shown) during the CGMD simulation at (A) $t = 0$ ns, (B) $t = 25$ ns, (C) $t = 250$ ns, (D) $t = 1000$ ns. Adapted from Paper III.
Figure 5.14: Anatase spherical nanoparticle ($r = 2$ nm) with the POPE bilayer (882 lipid molecules) during the CGMD simulation at (A) $t = 0$ ns, (B) $t = 75$ ns, (C) $t = 150$ ns, (D) $t = 1000$ ns. Na$^+$ and Cl$^-$ ions are shown as yellow and light green spheres. Adapted from Paper III.
5.3 Training a coarse-grained neural network potential on structural data (IV)

In the last part of the thesis work, we present an approach for training a transferable coarse-grained neural network potential (CG NNP) on structural data from higher-resolution simulations. We combine the ideas of the Behler-Parrinello NNPs [139; 140], which are used to construct efficient many-body atomic representations (symmetry functions), serving as the input to the neural network, and the ideas from the inverse Monte Carlo (IMC) method [178; 179] that provide the connection between the configurational averages and the parameters describing the potential. The resulting CG NNP can be used to perform Metropolis Monte Carlo (MC) simulations to sample configurational averages, but the computation of forces and running an MD simulation is also possible by taking the gradients of the total potential energy with respect to the CG coordinates (Equation 3.29). An important difference with the IMC method is that the proposed CG NNP can be trained to reproduce multiple reference systems at once, making the resulting model more transferable. Moreover, in a significant departure from the Behler-Parrinello NNP approach, the neural networks are not trained on multiple pairs of configuration-energy data, but instead learn the ensemble behavior from statistical averages. We test the method on a simple case of liquid argon which is described by Lennard-Jones (LJ) interactions, and then move on to a more complicated case of methanol-water solutions at CG resolution with implicit solvent. To run the MC simulations and the training, we develop a prototype software using the Julia programming language (version 1.8) [229] with the Flux machine learning library [230; 231], Zygote library for automatic differentiation [232] and Chemfiles [233] for reading and writing trajectory files. The software runs on multiple CPU cores to run parallel MC simulations and train on multiple reference datasets. The outline of the training algorithm for the proposed approach is shown in Figure 5.15.

On a more detailed level, the CG NNP training can be described as follows. We start with a CG system, for which the reference RDF is known from a higher-resolution simulation, for example from an all-atom molecular dynamics (AAMD) simulation. Using the Cartesian coordinates of the CG beads we compute the symmetry functions that encode the energetically relevant local environment of each CG bead. The symmetry functions are used as the input data to the CG NNP. Here we use only $G^2$ symmetry functions (Equation 3.26 and 3.27) with varying pa-
rameters of $\eta$ and $r_s$. The architecture of the neural network, specifically the number of neurons in each layer and the activation functions, has a profound effect on the capabilities of the CG NNP. Note that here we discuss only the simplest feedforward neural networks with full connectivity, however, other more advanced architectures exist. After a series of exhaustive searches for an optimal NN architecture, we have found two reasonable alternatives, which we name Model 1 and Model 2. The first one contains 8 input neurons, which means that it accepts 8 different symmetry function values from a single CG bead. It also has 3 hidden layers with 20 neurons each, and the ReLU activation function is used to ensure non-linear behavior. Model 2 has 24 input neurons and each hidden layer has 40 neurons instead of 20. Both models have a single output neuron, which contains the potential energy contribution of a given CG bead. Additionally, we test a linear modification of Model 1, which predicts the potential energy contribution from a linear combination of the input symmetry functions. Note that CG NNP is the same for all CG beads of the same type. Since our systems – the liquid argon and CG water-methanol mixtures, can be described by a single bead type, we need to train only one CG NNP per simulation. Moreover, in our CG NNP, we do not use any bias parameters – only weights are optimized. The starting values for the weights are set randomly following the Xavier
initialization method \[234\]. After computing the symmetry functions for each CG bead and forward propagating them through the CG NNPs, we sum all the predicted energy contributions into the total potential energy (Equation \[3.24\]). Having the ability to compute the total potential energy for a given CG configuration allows us to run an MC simulation and sample configurational properties, such as RDF. Each MC step involves randomly selecting a CG bead, its random displacement, and reevaluating the total potential energy. In our approach, we train such CG NNPs that minimize the difference between the RDF, sampled from the MC simulation, and the reference RDF from the higher-resolution simulations. To do that, we minimize the following loss function:

\[
L = \sum_{\alpha} \left( \left\langle g(r_{\alpha})^{\text{NNP}} \right\rangle - \left\langle g(r_{\alpha})^{\text{ref}} \right\rangle \right)^2
\]  \hspace{1cm} (5.2)

where \(\alpha\) is the index of an RDF bin, \(\left\langle g(r_{\alpha})^{\text{NNP}} \right\rangle\) is the RDF sampled with the CG NNP and \(\left\langle g(r_{\alpha})^{\text{ref}} \right\rangle\) is the reference RDF. Note that we compare the statistical averages of the RDFs. The next step is to compute the vector of all derivatives \(\frac{\partial L}{\partial w_i}\), where \(w_i\) is a weight parameter in the CG NNP:

\[
\frac{\partial L}{\partial w_i} = \sum_{\alpha} \frac{\partial L}{\partial \left\langle g(r_{\alpha})^{\text{NNP}} \right\rangle} \times \frac{\partial \left\langle g(r_{\alpha})^{\text{NNP}} \right\rangle}{\partial w_i}
\]  \hspace{1cm} (5.3)

While the first part of the derivative is trivial, the second one cannot be computed using the standard backpropagation method \[121\], as the RDF is not a direct output of the CG NNP. However, we can use an expression for such a derivative used in the IMC method (Equation \[4.6\]), which in the currently used notation would look like the following:

\[
\frac{\partial \left\langle g(r_{\alpha})^{\text{NNP}} \right\rangle}{\partial w_i} = -\beta \left( \left\langle \frac{\partial E}{\partial w_i} g(r_{\alpha})^{\text{NNP}} \right\rangle - \left\langle \frac{\partial E}{\partial w_i} \right\rangle \left\langle g(r_{\alpha})^{\text{NNP}} \right\rangle \right)
\]  \hspace{1cm} (5.4)

where both terms can be estimated during an MC simulation with the CG NNP, as the energy derivative \(\frac{\partial E}{\partial w_i}\) is an analytical function of the NNP parameters and the symmetry functions. Note that \(g(r_{\alpha})^{\text{NNP}}\) in Equation \[5.4\] is an estimator of the average RDF \(\left\langle g(r_{\alpha})^{\text{NNP}} \right\rangle\) for a specific configuration. After running an MC simulation with fixed NNP parameters, sampling the RDFs, and computing the loss gradients, we
can apply the gradients to the weight parameters and move on to the next training iteration with improved NNP parameters. The training iterations continue until the loss function is converged. To improve the transferability of the CG NNP, we perform the training iterations for multiple systems at once using the same CG NNP. For example, we can start the training for several water-methanol solutions with different methanol concentrations with the same randomly initialized CG NNP. We perform MC sampling for each system in parallel and compute the gradients. Then, the gradients are averaged and applied to all replicas of the CG NNP and the training continues with the same CG NNP in all different systems. This is equivalent to the minimization of a combined loss function that includes multiple RDFs. The resulting CG NNP learns to reproduce RDFs of different systems, becoming a more transferable CG model.

To improve the initial guess for the NNP parameters, we devised a special pre-training procedure that is based on reproducing the potential of mean force (PMF), recovered from the RDF:

\[ U(r) = -k_B T \ln \langle g(r) \rangle^{\text{ref}} \]  

(5.5)

However, we substitute the part of the PMF that results in discontinuities at short distances with a smooth linear repulsive potential:

\[
U(r_\alpha) = \begin{cases} 
-k_B T \ln \langle g(r_\alpha) \rangle^{\text{ref}} & \text{for } r_\alpha \geq r_0 \\
U(r_{\alpha_0}) + \Delta U \times (\alpha_0 - \alpha) & \text{for } r_\alpha < r_0 
\end{cases}
\] 

(5.6)

where \( r_0 \) is the shortest distance with non-zero RDF with index \( \alpha_0 \), and \( \Delta U = U(r_{\alpha_0}) - U(r_\alpha) \). The PMF, as shown in Equation (5.5), gives the correct potential only for an infinitely dilute system [177]. Still, it can serve as a reasonable first approximation for the CG potential, where the total energy is described as a sum of all pair interactions:

\[ E_{\text{PMF}} = \sum_{i<j} U(r_{ij}) \] 

(5.7)

Knowing an approximate value for the total potential energy allows the implementation of a training algorithm that is more similar to the Behler-Parrinello approach. During a pre-training step, we pick a random configuration from the reference trajectory at a CG resolution and evaluate the total potential energy with the NNP and the PMF. Then, we randomly displace a CG bead, as in a standard MC step, and evaluate
the total potential energy with both methods again. The goal is to minimize the differences between the computed potential energies obtained with both methods by minimizing the following loss function:

\[ L_{PMF} = (\Delta E^{PMF} - \Delta E^{NNP})^2 \] (5.8)

where the energy differences are computed for the initial CG configuration and the modified configuration with the displaced bead. Unlike the main training procedure, which includes the RDF sampling, the loss gradients are computed using the standard backpropagation [121] only:

\[
\frac{\partial L_{PMF}}{\partial w_i} = 2 (\Delta E^{PMF} - \Delta E^{NNP}) \times \left( \frac{\partial E^{NNP}_2}{\partial w_i} - \frac{\partial E^{NNP}_1}{\partial w_i} \right)
\] (5.9)

where \( E^{NNP}_2 \) is the total energy of the configuration after the displacement and \( E^{NNP}_1 \) is the total energy of the original configuration, both predicted by the NNP. The resulting gradients can be computed for several reference systems at once and their average is used to improve the NNP parameters. The pre-training is significantly faster than the main part of the training and provides a reasonable guess for the CG potential. Most importantly, after the pre-training the CG NNP learns to reproduce the repulsive part of the potential.

The liquid argon system which is used as a reference to test our new method is composed of 512 Ar atoms in 3D PBC with the density of \( \rho = 1374 \, \text{kg/m}^3 \) at \( T = 95 \, \text{K} \). We used the LJ potential to describe the interactions between the Ar atoms with the parameters by Rowley et al. [235]. To sample the reference RDF we run an MC simulation using our software. Then, we carry out MD simulations of the water-methanol systems with the General Amber Force Field (GAFF) [102, 236] with TIP3P water [214] using the GROMACS 2020.4 software package [187]. The composition of the water-methanol systems is shown in Table 5.8. At the CG resolution, a methanol molecule is described as a single bead located at the molecular center of mass, and the water molecules are excluded. The snapshots of the water-methanol systems at AA and CG resolutions are found in Figure 5.16. The molecular images are rendered using VMD software [221].

The training results and the sampled RDFs for the liquid argon system are shown in Figure 5.17. The loss steadily decreases until it converges and almost reaches zero. The RDF generated by the CG model after the pre-training correctly describes the repulsion region as well as...
Table 5.8: Composition of methanol-water systems in AAMD simulations. Adapted from Paper IV.

<table>
<thead>
<tr>
<th>N(CH₃OH)</th>
<th>N(H₂O)</th>
<th>Approx. CH₃OH conc., mol%</th>
<th>Box side, Å</th>
<th>Density, kg/m³</th>
</tr>
</thead>
<tbody>
<tr>
<td>512</td>
<td>0</td>
<td>100 %</td>
<td>32.3</td>
<td>808.2</td>
</tr>
<tr>
<td>512</td>
<td>57</td>
<td>90 %</td>
<td>32.8</td>
<td>819.5</td>
</tr>
<tr>
<td>512</td>
<td>128</td>
<td>80 %</td>
<td>33.4</td>
<td>830.7</td>
</tr>
<tr>
<td>512</td>
<td>220</td>
<td>70 %</td>
<td>34.2</td>
<td>843.3</td>
</tr>
<tr>
<td>512</td>
<td>340</td>
<td>60 %</td>
<td>35.2</td>
<td>856.4</td>
</tr>
<tr>
<td>512</td>
<td>512</td>
<td>50 %</td>
<td>36.5</td>
<td>871.2</td>
</tr>
<tr>
<td>512</td>
<td>768</td>
<td>40 %</td>
<td>38.4</td>
<td>887.7</td>
</tr>
<tr>
<td>512</td>
<td>1195</td>
<td>30 %</td>
<td>41.1</td>
<td>906.6</td>
</tr>
<tr>
<td>512</td>
<td>2048</td>
<td>20 %</td>
<td>45.7</td>
<td>928.7</td>
</tr>
<tr>
<td>512</td>
<td>4608</td>
<td>10 %</td>
<td>55.6</td>
<td>954.6</td>
</tr>
</tbody>
</table>

gives the correct position of the first peak. The refined potential provides almost exactly the same RDF, as the reference obtained with the LJ potential. For this simple test case, we have used a linear modification of Model 1, which contains only 8 tunable parameters (one weight parameter for each symmetry function value). To train a transferable water-methanol CG NNP, we run the training with four systems in parallel – containing 10, 40, 60, and 100 mol% of methanol. The training is performed for both Models 1 and 2, but the results are additionally compared with the linear model. Figure 5.18 shows the convergence of the loss function for all three models. Neither of the models can reach loss values close to zero, as in the liquid argon case, as now the CG NNP tries to reproduce several systems at once. The largest loss reduction is observed for Model 2, but Model 1 reaches a similarly low value around iteration 25. Even though the linear model manages to minimize the loss better than we initially expected, it remains consistently higher than both CG NNPs.

After the training, we validate the models by running MC simulations of the water-methanol mixtures of all compositions for which the reference data is available using the obtained CG NNPs and compare the resulting RDFs with the reference ones. Additionally, we compare the distribution of angles for all molecular triplets in the system (with a cut-off radius of 6.2 Å). The results for Model 1 are shown in Figures 5.19 and 5.20. Furthermore, we compare the results predicted by our models with those obtained for CG simulations with the IMC potentials trained on different methanol concentrations. The data is shown in Tables 5.9 and 5.10.
**Figure 5.16:** AA and CG representations of methanol-water mixtures. Pure methanol at the AA resolution (A) and the CG resolution (C), methanol-water mixture (20 mol% CH$_3$OH) at the AA resolution (B), and the CG resolution (D). Adapted from Paper IV.
Figure 5.17: Loss convergence for the liquid argon system (left) and the comparison of the reference RDF and the RDF sampled with the linear NNP. Adapted from Paper IV.

Figure 5.18: Loss convergence for the CG methanol models. Adapted from Paper IV.
Figure 5.19: Comparison of the reference RDF and the RDF sampled with Model 1 for CG water-methanol mixtures. Adapted from Paper IV.

Figure 5.20: Comparison of the reference angular distribution and the distribution sampled with Model 1 for CG water-methanol mixtures. Adapted from Paper IV.
Table 5.9: Comparison of the RDF losses from different methanol CG models. IMC-X – IMC CG potential trained on a methanol-water system with X mol% of methanol. Adapted from Paper IV.

<table>
<thead>
<tr>
<th>%mol</th>
<th>Linear</th>
<th>Model 1</th>
<th>Model 2</th>
<th>IMC-100</th>
<th>IMC-80</th>
<th>IMC-60</th>
<th>IMC-40</th>
<th>IMC-20</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.7738</td>
<td>0.6253</td>
<td>0.5396</td>
<td>0.0002</td>
<td>0.0997</td>
<td>0.4200</td>
<td>1.0049</td>
<td>2.2086</td>
</tr>
<tr>
<td>90</td>
<td>0.5334</td>
<td>0.4179</td>
<td>0.3357</td>
<td>0.0603</td>
<td>0.0347</td>
<td>0.2176</td>
<td>0.6672</td>
<td>1.6578</td>
</tr>
<tr>
<td>80</td>
<td>0.3613</td>
<td>0.2860</td>
<td>0.2109</td>
<td>0.0925</td>
<td>0.0001</td>
<td>0.1071</td>
<td>0.4555</td>
<td>1.2910</td>
</tr>
<tr>
<td>70</td>
<td>0.2444</td>
<td>0.1947</td>
<td>0.1298</td>
<td>0.2175</td>
<td>0.0270</td>
<td>0.0266</td>
<td>0.2697</td>
<td>0.9288</td>
</tr>
<tr>
<td>60</td>
<td>0.1543</td>
<td>0.1315</td>
<td>0.0676</td>
<td>0.3996</td>
<td>0.1038</td>
<td>0.0002</td>
<td>0.1381</td>
<td>0.6557</td>
</tr>
<tr>
<td>50</td>
<td>0.1109</td>
<td>0.1110</td>
<td>0.0425</td>
<td>0.7553</td>
<td>0.2909</td>
<td>0.0438</td>
<td>0.399</td>
<td>0.4097</td>
</tr>
<tr>
<td>40</td>
<td>0.1097</td>
<td>0.1295</td>
<td>0.0472</td>
<td>1.5551</td>
<td>0.7374</td>
<td>0.2255</td>
<td>0.0003</td>
<td>0.2022</td>
</tr>
<tr>
<td>30</td>
<td>0.1769</td>
<td>0.2007</td>
<td>0.1089</td>
<td>3.1430</td>
<td>1.5926</td>
<td>0.6116</td>
<td>0.0506</td>
<td>0.0582</td>
</tr>
<tr>
<td>20</td>
<td>0.3681</td>
<td>0.3638</td>
<td>0.2490</td>
<td>6.2870</td>
<td>3.1816</td>
<td>1.3150</td>
<td>0.2383</td>
<td>0.0006</td>
</tr>
<tr>
<td>10</td>
<td>0.7369</td>
<td>0.6429</td>
<td>0.5396</td>
<td>11.7863</td>
<td>5.7367</td>
<td>2.4490</td>
<td>0.6468</td>
<td>0.0872</td>
</tr>
</tbody>
</table>

Mean loss 0.3572 0.3103 0.2221 2.4295 1.1805 0.5416 0.3511 0.7500
Min loss 0.1097 0.1110 0.0425 0.0002 0.0001 0.0002 0.0003 0.0006
Max loss 0.7738 0.6429 0.5396 11.7863 5.7367 2.4490 0.6468 0.0872

Table 5.10: Comparison of the angle distribution losses from different methanol CG models. IMC-X – IMC CG potential trained on a methanol-water system with X mol% of methanol. Adapted from Paper IV.

<table>
<thead>
<tr>
<th>%mol</th>
<th>Linear</th>
<th>Model 1</th>
<th>Model 2</th>
<th>IMC-100</th>
<th>IMC-80</th>
<th>IMC-60</th>
<th>IMC-40</th>
<th>IMC-20</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.0532</td>
<td>0.0382</td>
<td>0.0482</td>
<td>0.0040</td>
<td>0.0117</td>
<td>0.0333</td>
<td>0.0673</td>
<td>0.1367</td>
</tr>
<tr>
<td>90</td>
<td>0.0387</td>
<td>0.0328</td>
<td>0.0388</td>
<td>0.0104</td>
<td>0.0120</td>
<td>0.0192</td>
<td>0.0500</td>
<td>0.1056</td>
</tr>
<tr>
<td>80</td>
<td>0.0312</td>
<td>0.0222</td>
<td>0.0278</td>
<td>0.0065</td>
<td>0.0040</td>
<td>0.0157</td>
<td>0.0352</td>
<td>0.1034</td>
</tr>
<tr>
<td>70</td>
<td>0.0230</td>
<td>0.0221</td>
<td>0.0236</td>
<td>0.0169</td>
<td>0.0052</td>
<td>0.0062</td>
<td>0.0311</td>
<td>0.0708</td>
</tr>
<tr>
<td>60</td>
<td>0.0152</td>
<td>0.0185</td>
<td>0.0207</td>
<td>0.0298</td>
<td>0.0096</td>
<td>0.0038</td>
<td>0.0172</td>
<td>0.0544</td>
</tr>
<tr>
<td>50</td>
<td>0.0138</td>
<td>0.0134</td>
<td>0.0145</td>
<td>0.0665</td>
<td>0.0196</td>
<td>0.0023</td>
<td>0.0121</td>
<td>0.0389</td>
</tr>
<tr>
<td>40</td>
<td>0.0092</td>
<td>0.0054</td>
<td>0.0075</td>
<td>0.1619</td>
<td>0.0612</td>
<td>0.0160</td>
<td>0.0037</td>
<td>0.0224</td>
</tr>
<tr>
<td>30</td>
<td>0.0058</td>
<td>0.0086</td>
<td>0.0055</td>
<td>0.3617</td>
<td>0.1600</td>
<td>0.0482</td>
<td>0.0062</td>
<td>0.0112</td>
</tr>
<tr>
<td>20</td>
<td>0.0103</td>
<td>0.0135</td>
<td>0.0097</td>
<td>0.7934</td>
<td>0.2533</td>
<td>0.0855</td>
<td>0.0119</td>
<td>0.0504</td>
</tr>
<tr>
<td>10</td>
<td>0.0261</td>
<td>0.0257</td>
<td>0.0388</td>
<td>1.2850</td>
<td>0.5930</td>
<td>0.1864</td>
<td>0.0381</td>
<td>0.0163</td>
</tr>
</tbody>
</table>

Mean loss 0.0227 0.0200 0.0235 0.2736 0.1130 0.0417 0.0273 0.0565
Min loss 0.0058 0.0054 0.0055 0.0040 0.0040 0.0023 0.0037 0.0054
Max loss 0.0532 0.0382 0.0482 1.2850 0.5930 0.1864 0.0673 0.1367

Both figures illustrate how the CG NNP successfully reproduces the RDFs for the systems outside of the training set and even provides correct angular distributions, even though they were not used to train the CG NNP. Our CG NNPs compare favorably to the IMC potentials, as demonstrated by the tabular data. Both Models 1 and 2 provide lower mean and maximum loss values compared to the IMC potentials. The latter potentials produce extremely accurate results for the concentration they were trained on, however, the error gradually increases as the concentration changes. We hypothesize that a reason behind a good reproduction of the RDFs and angular distributions across all water-methanol systems by our CG NNPs is that they explicitly describe many-body interactions, as illustrated by Figure 5.21. The potential energy surface...
of three methanol molecules (of which two are static) predicted by our CG NNP contains a non-zero remainder after subtracting all pair potential contributions. The three-body contribution enhances the attraction close to the contact distance of around 3-4 Å and features an angular dependence. Moreover, when the two static molecules are close to each other, a repulsive three-body component becomes prominent (subfigure B and C), but it disappears when the static molecules are separated (subfigure D).

Our new method shows that it is possible to train a CG NNP on structural data from multiple reference systems and obtain a transferable model that learns many-body interactions on the example of water-methanol solutions. It does not require a large training set to learn the interactions, as long as the reference RDF is sufficiently smooth and representative of the given system. Moreover, the current method can be used to fit the potential from ab initio data and even from the experimentally measured RDFs. The transferability over varying compositions becomes extremely important in systems with phase separations, particularly where a hydrophobic effect plays a significant role. Expanding the method to multiple CG bead types enables the simulation of increasingly more complicated systems, eventually reaching the level of CG lipids and potentially even the surface-lipid interfaces. Future studies of the new method as well as its development are necessary to reach the level of such complicated systems. We note that the current computational performance of the method is quite limited, as it takes roughly 5-10 times as long to run an MC simulation with the CG NNP when compared to a pair potential. However, performance improvements can be achieved by software optimization as well as by finding a more efficient structural descriptor and neural network architecture.
Figure 5.21: An illustration of the three-body component of the NNP trained for the non-linear CG methanol model (Model 1). (A): a pair component of the NNP computed as the energy of two CG particles; (B-D): three-body component of the NNP computed as the energy difference $\Delta E^{(3)} = E^{(3)}(1,2,3) - E^{(2)}(1,2) - E^{(2)}(2,3) - E^{(2)}(1,2)$, where $E^{(3)}(1,2,3)$ is the energy of three CG particles, and $E^{(2)}(i,j)$ is the energy of each of the three particle pairs. The three-body component is shown as a density map of the position of the third particle when the first two particles are fixed at distances 3.4, 4.5, and 6.3 Å respectively. Adapted from Paper IV.
6. Conclusions and outlook

6.1 Conclusions

This thesis work demonstrates how systematic coarse-graining methods can be applied to simulate interfaces consisting of inorganic nanomaterials and biological molecules, which has significant relevance in various research areas, most notably nanotoxicity, but additionally – drug design, enzyme immobilization, and hybrid biomaterials. The atomistic simulations of lipids near TiO$_2$ surfaces and nanoparticles, shown in Papers I and II, revealed that a complex interplay of the surface composition, curvature, lipid functional groups, and solvent effects governs the overall interactions of the surface with the biomolecules. However, the scales of the related molecular events are often incompatible with the current computational capabilities. Thus, there is a need for efficient, low-cost, and large-scale molecular models that can capture the complexities of such bio-nano interfaces. In Paper III, by using the inverse Monte Carlo coarse-graining approach together with the low-resolution mappings of the lipid molecules and the TiO$_2$ surfaces, we obtained coarse-grained potentials that accurately reproduce the structural properties of the higher-resolution reference systems. After the validation of the resulting coarse-grained models against the atomistic data, we tested our models on a variety of large-scale systems containing TiO$_2$ nanoparticles and lipid molecules, predicting how the lipid adsorption changes with the size of the nanoparticle, as well as simulating a partial wrapping of a small anatase nanoparticle. Furthermore, in Paper IV, we developed a new coarse-graining method employing neural network potentials trained on structural data and tested it on a simple system of water-methanol solutions of different concentrations. By training our coarse-grained neural network potentials on multiple reference systems at once, we achieved a higher degree of transferability when compared to the pair coarse-grained potentials optimized with the inverse Monte Carlo method. We further demonstrated that the coarse-grained neural network potential acts as a true many-body potential, which can explain its ability to reproduce high-order correlations and better transferability.
6.2 Further developments

The further developments of the current work may primarily consist of two main directions – the expansion and the refinement of the coarse-grained model of the lipid molecules with TiO$_2$ surfaces, and the optimization of the coarse-grained neural network approach.

First of all, in our coarse-grained models, the nanoparticles should be represented more realistically, with separate faces and defect sites, which can potentially act as specific anchoring points to the biological molecules. Then, the variety of shapes of the nanoparticles can be expanded to study their effect on the interactions with biomolecules. Finally, by carrying out more inverse Monte Carlo simulations of different surface-biomolecule pairs, we can enrich the composition of the simulated biological membranes, adding more lipid types, but also sugars, proteins, and other biological molecules. It allows simulations of more complex and realistic biological systems, potentially uncovering more mechanisms of the nanomaterial action.

Although promising, our new coarse-graining approach based on the neural network potential is currently in the early stages of development. There is a crucial need for the optimization of the structural descriptors used to represent the local environment of each coarse-grained particle, the neural network architecture as well as the software performance. Moreover, the method should be tested on more complex systems, beyond the isotropic fluids. A step further could be a reproduction of the hydrophobic effect in water-methane systems and a simulation of simplistic coarse-grained models of lipids. But one of the main challenges in this endeavor is certainly to strike a good balance between the predictive quality of the model, its transferability, and its performance.
Acknowledgments

First of all, I want to thank Sasha, my main supervisor. Уже после первой нашей личной встречи, в Петергофе, я знал что мне очень повезло с научным руководителем. Не только вы обладаете большим научным опытом и знаниями, но также и являетесь прекрасным человеком, который заботится о людях вокруг, помогает своим студентам, аспирантам и другим коллегам. Я всегда особенно ценил вашу отзывчивость, ваше участие, и доверие ко мне. Спасибо вам огромное за возможность работать в вашей группе!

Then, I want to leave a few words of appreciation to my fellow group members as well as the former ones. It was so great to work with all of you! Fredrik, you’ve been with me from the very beginning of my PhD and I thank you for sharing your experience with me, for your hospitality, and for the fun time we had together – having a fika, taking a walk, and a swim. Roja, thank you for the amazing parties, dances, and spontaneous tea gatherings! Radost, it was so much fun to play cello with you in the orchestra – now I know that I will play it more in the future. Максим, я очень рад, что ты присоединился к нашей группе, спасибо тебе за помощь с нашим проектом, ты качественно улучшил его, и работать над ним с тобой стало гораздо приятнее и интереснее! Amanda, it brings me joy when such smart people like you do computer simulations and grow to like it. I wish you all the best in your future studies! Дима, я рад что успел застать тебя до моего ухода, желаю тебе дальнейших успехов и всего хорошего! Bojan, thank you for helping me with the teaching and the beautiful painting you left in our office. Erik, your work and experience have been of particular help to me at the beginning of my PhD, and I thank you for that. Marzieh, thank you for organizing the group activities that brought our group together. Federico, thank you for the teaching materials and the PhD thesis template, that was of great help to me! Инна, спасибо тебе за помощь в начале моей аспирантуры, твоя поддержка очень помогла мне освоиться в новой атмосфере. Nikos, you too were a supportive office mate, helping me with Linux and the work in general. Zoey, even though I have not participated in your supervision as much as Fredrik, I’m still
happy that I have contributed to your nice work. Lorenzo, thanks for the nice discussions about titanium dioxide and high-performance computing. Inna, it was great to have you as a guest researcher in our group! Samuel, even though I have not met you in person, you’ve managed to contribute significantly to my work, helping me to build more realistic representations of the TiO$_2$ surfaces. Маша, было очень здорово что ты приехала к нам в Стокгольм и что мы снова работали вместе! Саша, спасибо тебе за разработку MagiC’а и за помощь в получении грубозернистых моделей – это сделало мои результаты реальностью!

In addition, I want to thank the staff at MMK. Arnold, for being my co-supervisor, helping me with the studies, organizing my performance interviews, and telling the travel stories over lunches, Baltzar, for being close to our group and participating in the meetings, Mattias, for providing me with helpful feedback for my thesis, Niklas and Zoltan, for helping me with the teaching at the Soft Matter course, Aleks, for the help with the data analysis course, Ulrich and Leopold, for the interesting collaborations within MMK, Gunnar, for the help during my time in the PhD council, Mirva, for covering for me once as the exam guard, but also Ulrika, Volodymyr, Mats, Anne, Karim, Ulrike, Lesya, Alla and Serhii. Big thanks also go to the administrative staff – to Tatiana, for the warm welcome to MMK, for the help with both administrative and personal matters, Ann-Sofi, Elin, Åsa, Lilian, Daniel, Annette, Jenny, Camilla, for handling the contracts, salaries, documents, reimbursements and more, Helmi, for the help with Ladok, Daniel, for the travel-related questions, Roffe and Erik, for the IT administration and the help with the computer hardware, Christer, for handling the access cards, and to the KÖL staff – Lilli, Lina and Jakob. Furthermore, I want to thank my teachers and mentors during my time at MMK – Michael Odelius, Jonatan Öström, Berk Hess, Jonas Lindemann, Stefano Markidis, Ribana Roscher, Inga Koszalka, Olav Vahtras, Staffan Bergwik, Emily Freeland and Erik Tengstrand. I want to additionally thank my external collaborators – Boštjan, Hana, Aleksandar, Janez, Vladimir, Wojtek, Natalia, Iain, and of course, the National Academic Infrastructure for Supercomputing in Sweden (NAISS) for providing the resources for my simulations, the Swedish Research Council, Stockholm University Åkesson Foundation, EU Horizon 2020 SmartNanoTox and NanoSolveIt projects for the funding, as well as Stockholm University Library and US-AB for the printing services and the dissertation support.
My time as a PhD student at Stockholm University would not be the same and as fun thanks to the other people I've met along the way – Paulo, Pierre, Zé, Alisa, Aditya, Wassilios, Milla, Rihards, Martin, Kosta, Jedrzej, Sonya, Zhenya, Ruslan, Laura, Ehsan, Micke, Elnaz, Brando, Yulia, Stefanie, Cody, Ioannis, Emily, Tanya, Egor, Vika, Artyom, Masha, Dima. I also want to thank the PhD students who were a part of the PhD council together with me – Molly, Fredrik, Ira, Dimi, Atefeh, Vahid, Josefine, Sadaf, Natalia, Carina, Roja, Tim.

Я хочу также поблагодарить членов группы компьютерного моделирования Института Химии СПбГУ, в которой я работал со второго курса бакалавриата до окончания магистратуры и приобрёл бесценный опыт (и просто хорошо провёл время!) – Сизова В.В., Сизову А.А., Ванина А.А., Готлиба И.Ю., Бродскую Е.Н., Машу, Илью, Лену, Соню, Пащу и Настию. Кроме того я хочу поблагодарить Кудрева А.Г. за интересный проект, над которым мне почастивилось работать в магистратуре, а также Викторова А.И. и Тимошкина А.Ю. за специальные курсы и рекомендательные письма перед поступлением в аспирантуру.

Хочу сказать спасибо и ребятам, некоторых из которых я знаю очень давно, которые были рядом – Саше, Ване, Серёже, Родиону, Артёму, Альфредо, Илье, Васе, Косте, Андрею, Жене, Вике, Вите, Ксюше, Юре, Насте, Коле и Севе.

Мой переезд в Швецию, моё образование, и многое другое не было бы возможным без моей семьи – папы, мамы, Димы, Илюшки, Марка и Иры. Также хочу упомянуть бабушку, тёту Лену и Валеру, Любую, Катю, тётю Иру, Лёву, Пащу и дядю Женю, с которыми я бы хотел видеться чаще. Благодарности заслуживают и наши с Ирой кошки – Пуша и Лима, которые принесли нам столько радости и умилений.

И наконец, я хочу сказать отдельное и огромное спасибо моей дорогой жене Ире. Я мог бы многое написать здесь, но оставлю одно – ты наполнила мою жизнь смыслом!
References


Roger W. Hockney and James W. Eastwood. **Computer Simulation Using Particles.** CRC Press, 2021. 35


Michael A. Nielsen. **Neural Networks and Deep Learning** 2018. 37

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. **Multilayer Feedforward Networks are Universal Approximators** [Neural Netw., 2(5):359–366, 1989.]


Seppo Linnainmaa. **Taylor Expansion of the Accumulated Rounding Error** [BIT, 16(2):146–160, 1976. 38]


