Applications of Neural Networks in Theory and Simulation of Slow Dynamics

Inaugural-Dissertation

zur Erlangung des Doktorgrades

der Mathematisch-Naturwissenschaftlichen Fakultät

der Heinrich-Heine-Universität Düsseldorf

vorgelegt von

Leon Frederik Granz

aus Mettmann

Köln, Januar 2026

aus dem Institut für Theoretische Physik II: Weiche Materie
der Heinrich-Heine-Universität Düsseldorf

Gedruckt mit der Genehmigung der
Mathematisch-Naturwissenschaftlichen Fakultät der
Heinrich-Heine-Universität Düsseldorf

Berichterstatter:

1. Prof. Dr. Thomas Voigtmann
2. Prof. Dr. Jürgen Horbach

Tag der mündlichen Prüfung:

08. April 2026

Eidesstattliche Versicherung

Ich, Leon Frederik Granz, versichere an Eides Statt, dass die Dissertation von mir selbständig und ohne unzulässige fremde Hilfe unter Beachtung der „Grundsätze zur Sicherung guter wissenschaftlicher Praxis an der Heinrich-Heine-Universität Düsseldorf“ erstellt worden ist.

Ort, Datum	L. F. Granz

Hinweis zur Verwendung von KI-Werkzeugen

Bei der Anfertigung dieser Dissertation wurden generative KI-Werkzeuge unterstützend eingesetzt. Dies umfasste sprachliche Überarbeitungen und Umformulierungen von Text sowie technische Unterstützung bei der Erstellung und Optimierung von Code (z. B. LaTeX-Quelltext und Python-Skripte).

Abstract

This thesis applies neural network methods to two types of problems in statistical and computational physics. The first concerns the construction of neural network potentials (NNPs) for particle-based simulations. The second involves the development of a neural network inverse Laplace transform (NNLT) that enables the calculation of memory kernels for slow dynamics in the context of generalized Langevin equations and, in particular, the mode-coupling theory of the glass transition (MCT).

In the first part, NNPs are evaluated on model systems of increasing complexity, including glass-forming binary mixtures. For Lennard-Jones systems and Kob–Andersen mixtures, trained NNPs reproduce structural and dynamical observables with high accuracy when the relevant regions of phase space are sampled. The Voronoi potential, based on a spatial tessellation, is then used as a demanding test case for applying NNPs to simulations of glassy dynamics. Different architectures are compared, with graph-based models outperforming simpler approaches and delivering accurate energies and forces. These models also enable stable simulations even in regimes with slow relaxation. Extensions to binary Voronoi mixtures reveal limitations as the system approaches dynamical arrest. The resulting training strategies transfer directly to elemental boron, where NNPs are trained on density-functional-theory data.

The second part introduces the NNLT, which is designed to address the ill-conditioned inverse Laplace transform of correlation functions that arises in MCT. Trained on superpositions of decaying exponentials, the NNLT generalizes to Laplace-domain correlation functions from both MCT and Brownian dynamics. Combined with the Laplace-domain Mori–Zwanzig equation, it yields accurate time-domain memory kernels. An iterative normalization scheme, refined using hydrodynamic scaling arguments, stabilizes the inversion and enables the calculation of memory kernels directly from simulation data. Comparisons of simulations with MCT reveal systematic deviations near dynamical arrest. Using the computed kernels, neural network functionals are constructed that approximate the memory kernel of the MCT-type generalized Langevin equation.

Overall, the results demonstrate that neural networks enhance theoretical and computational studies of many-body systems. Neural network potentials provide flexible models of potential energy surfaces, while the NNLT provides access to dynamical quantities that are otherwise numerically inaccessible. Together, these methods show how machine learning extends and complements established approaches in the study of complex fluids and glassy dynamics.

List of Acronyms

ADAMW	Adaptive Moment Estimation with Weight Decay
AGL	Adaptive Group Lasso
AIMD	Ab Initio Molecular Dynamics
BD	Brownian Dynamics
DeePMD	Deep Potential Molecular Dynamics
DFT	Density Functional Theory
EDBD	Event-Driven Brownian Dynamics
FMT	Fundamental-Measure Theory
FT	Fourier Transform
HDNNP	High-Dimensional Neural Network Potential
ISF	Intermediate Scattering Function $S(q,t)$
KA	Kob–Andersen
LAMMPS	Large-Scale Atomic/Molecular Massively Parallel Simulator
LJ	Lennard-Jones
LT	Laplace Transform
MCT	Mode Coupling Theory of the Glass Transition
MD	Molecular Dynamics
MSD	Mean Squared Displacement $\langle \Delta r^2(t) \rangle$
MSE	Mean Squared Error
MZ	Mori–Zwanzig (Equation)
n2p2	Neural Network Potential Package
NequIP	Neural Equivariant Interatomic Potential
NN	Neural Network
NNLT	Neural Network (Inverse) Laplace Transform
NNP	Neural Network Potential
NVE	Microcanonical Ensemble
NVT	Canonical Ensemble
PES	Potential Energy Surface

RDF Radial Distribution Function $g(r)$

RMSE Root Mean Squared Error

SF Static Structure Factor $S(q)$

SISF Self-Intermediate Scattering Function $S^{\mathrm{self}}(q,t)$

Contents

1. Introduction

Artificial intelligence, machine learning, and neural networks have a long and varied history, with conceptual roots in early developments of statistics and computer science. However, over the last decade, they have experienced an unprecedented surge in widespread adoption in disciplines ranging from the natural sciences to engineering, medicine, and even the humanities [1–3]. This rapid expansion has in part been fueled by strong public and commercial interests, often amplified by the rather unfortunate terminology suggesting notions of sentience or genuine, human-like "intelligence". Nevertheless, the scientific community has benefitted greatly from the resulting advances in machine learning, and deep learning techniques in particular have opened entirely new avenues for data analysis and interpretation [4, 5].

By leveraging enormous computational resources, and utilizing architectural breakthroughs, neural networks (NNs) routinely succeed in areas where conventional methods stagnate. One major feature underlying these accomplishments is the property of NNs to act as universal function approximators [6], i.e., given an adequate architecture and sufficiently diverse training data, they are capable of finding highly non-linear and complex mappings between inputs and outputs. Unfortunately, and despite their effectiveness, the internal mechanisms of NNs are difficult to interpret. Although the mathematical foundations of their approximation capabilities are well established [4, 5, 7], the precise nature of what is represented and the mechanisms by which relevant features are encoded remain active research topics.

Neural Network Potentials

One area where neural networks have advanced the state of the art is particle-based simulations. Neural network potentials (NNPs) provide a mapping between atomic configurations and their potential-energy surface (PES), allowing an approximation of both energies and forces in many-body systems [8–17]. This enables molecular dynamics simulations that were previously computationally infeasible. Such methods are applied to complex materials, where they extend the reach of simulations beyond the limits of simple empirical potentials or expensive quantum models. Classical pair potentials are computationally efficient, but often too limited to capture physical phenomena accurately. On the other

hand, quantum-mechanical methods such as density functional theory (DFT) offer high accuracy but are restricted to comparatively small system sizes due to their computational cost [18]. NNPs attempt to bridge this gap, combining the scalability of empirical potentials with the accuracy of first-principles approaches.

In this thesis, NNPs are first benchmarked on Lennard-Jones systems [19–21] before being applied to the Voronoi potential [22, 23], an unconventional model to simulate the dynamics of biological cells. For both systems, binary variants are considered, providing access to the physics of glassy dynamics and slow relaxations. These models provide a test of the generality and applicability of NNPs, allowing the investigation of training procedures without generating expensive ab initio datasets. Additionally, by working with analytically exact potentials, the comparison between true interactions and their neural network approximations becomes more transparent, offering a rare opportunity to partially "look inside the black box" of NNs. Finally, an excursion to elemental boron demonstrates the relevance of the developed methodologies, showing improvements over existing NNPs.

Neural Network inverse Laplace Transform

Beyond modeling PES, neural networks can also be applied to capture numerically intractable functional relationships found in theoretical physics. A particularly challenging example arises in the mode coupling theory of the glass transition (MCT), where correlation functions are determined from the Mori–Zwanzig (MZ) equation by approximating the memory kernel as a functional of the correlation function [24–26]. Inverting this relation, i.e., computing an unknown memory kernel from a known correlation function, requires performing an inverse Laplace transform (LT), which is well known to be numerically ill-conditioned [27–29].

To address this challenge, this thesis introduces the neural network inverse Laplace transform (NNLT), in which a NN is used to approximate the inverse LT of correlation functions. Once trained successfully, the network can directly calculate time-domain functions from their Laplace-domain representations. This makes it possible to calculate memory kernels from correlation functions via the Laplace-domain MZ equation, both within MCT and for simulation data, such as those generated from Brownian dynamics simulations. The resulting memory kernels allow for direct comparisons between theory and simulations, opening new routes for validating and expanding MCT.

Furthermore, with access to pairs of correlation functions and corresponding memory kernels for simulations of Brownian particles, it becomes feasible to search for a functional relation similar to the mode coupling approximation. Neural networks can again be employed for this purpose, and even the demonstration that such an approximate mapping exists provides new insights into the connection between MCT and computer simulations.

Outline

The thesis is structured as follows.

Chapter 2 introduces some general methods relevant to this work, including particle-based computer simulations and the analysis of many-particle systems, some basic concepts of neural networks, and the Laplace transform.

Chapter 3 focuses on neural network potentials. After introducing architectures, describing their construction, and elaborating the training procedure, NNPs are employed for the Lennard-Jones potential as a benchmark system, and subsequently to the Voronoi potential as a geometrically challenging test case. Binary systems are considered to investigate performance for systems with slow relaxations and close to dynamical arrest. An application to elemental boron demonstrates how methodologies developed with computationally accessible models can be transferred to more challenging systems.

Chapter 4 introduces the neural network inverse Laplace transform to ultimately calculate memory kernels for Brownian particles. The neural network is trained on synthetic correlation functions and subsequently validated on data from both MCT and simulations. The NNLT is then incorporated into a framework to compute time-domain memory kernels from their Laplace-domain represenation obtained via the Mori–Zwanzig equation. Following the validation of the framework with data from MCT, it is applied to compute memory kernels for Brownian dynamics simulations. Finally, based on the resulting pairs of correlation functions and memory kernels, a neural network functional mimicking the mode coupling approximation is investigated. This establishes a novel approach to connect theory and simulation.

2. Methods

This chapter provides a short overview of some methods employed in this thesis. Particle-based simulations are introduced as a means to study many-particle systems by generating trajectories from which structural and dynamical observables can be obtained. Molecular dynamics and Brownian dynamics are considered. A brief introduction to neural networks is given, highlighting their role in data analysis and function approximation. Finally, due to its relevance to mode coupling theory, the Laplace transform is presented, including a method for numerical evaluation.

2.1 Computer Simulations of Many-Particle Systems

A variety of simulation techniques can be employed to study the microscopic dynamics of an interacting many-particle system, tailored to the given underlying physical considerations. Molecular dynamics (MD) simulations model systems with given interaction potentials, while event-driven Brownian dynamics (EDBD) simulations are used for hard-disk or hard-sphere systems obeying Langevin dynamics. While MD integrates Newton's equations of motion with forces derived from interparticle potentials, EDBD combines stochastic updates with exact handling of elastic collisions, thereby providing an efficient representation of overdamped Brownian motion. Both approaches generate trajectories of particle positions and, when applicable, velocities, from which structural and dynamical observables can be calculated.

2.1.1 Molecular Dynamics Simulation

Molecular dynamics (MD) simulations provide a framework to investigate the microscopic dynamics of atomic and molecular systems [30, 31]. A system of N atoms without relevant internal degrees of freedom in d spatial dimensions is fully described at time t by the positions $\boldsymbol{r}_i(t)$ and velocities $\boldsymbol{v}_i(t)$ of all atoms i, each with mass m_i. The temporal evolution of these quantities follows from Newton's equations of motion, with forces $\boldsymbol{F}_i$ obtained as the gradients of an interaction potential V. Depending on the complexity of

the system under consideration, this potential may range from simple pair interactions to more elaborate empirical many-body models, or it may be derived directly from first-principles methods such as density functional theory (DFT). In recent years, machine learning potentials have emerged, which try to achieve DFT-level accuracy at a fraction of computational cost.

Once forces are defined, the integration of Newton's equations yields the trajectories for all particles. Among the available numerical schemes, the Velocity-Verlet algorithm [32, 33] is widely employed, balancing accuracy and efficiency. It updates velocities and positions in a staggered manner,

$$\boldsymbol{v}_i(t + \tfrac{1}{2}\Delta t) = \boldsymbol{v}_i(t) + \tfrac{1}{2}\boldsymbol{a}_i(t)\Delta t, \tag{2.1}$$

$$\boldsymbol{r}_i(t + \Delta t) = \boldsymbol{r}_i(t) + \boldsymbol{v}_i(t + \tfrac{1}{2}\Delta t)\Delta t, \tag{2.2}$$

$$m_i\boldsymbol{a}_i(t + \Delta t) = \boldsymbol{F}_i(\boldsymbol{r}_i(t + \Delta t)), \tag{2.3}$$

$$\boldsymbol{v}_i(t + \Delta t) = \boldsymbol{v}_i(t + \tfrac{1}{2}\Delta t) + \tfrac{1}{2}\boldsymbol{a}_i(t + \Delta t)\Delta t, \tag{2.4}$$

where the equations of motion are discretized with a timestep Δt.

The system is typically confined within a simulation box. Boundary conditions are chosen to reflect the physical situation of interest. Fixed, repulsive walls can mimic confinement, while periodic boundary conditions are most commonly applied to represent bulk phases. In the latter case, the system is effectively replicated in all spatial directions, allowing particles to interact with images of their neighbors and minimizing surface effects.

MD simulations are performed in statistical ensembles. In the microcanonical ensemble (NVE), the number of particles N, volume of the simulation box V, and total energy E are conserved, following strictly Newton's equation of motion. Alternatively, canonical simulations (NVT) enforce a fixed temperature T, often through thermostatting schemes such as the Nosé-Hoover thermostat [34, 35]. While thermostats introduce an unphysical coupling, NVT simulations are particularly valuable during equilibration, ensuring the system reaches the desired temperature before switching to an NVE ensemble for production runs.

From trajectories, both static and dynamic properties can be calculated within statistical mechanics. For example, the instantaneous temperature is obtained microscopically as

$$T = \frac{2}{dN}E_{\text{kin}}, \tag{2.5}$$

$$E_{\text{kin}} = \frac{1}{2}\sum_i^N m_i\boldsymbol{v}_i^2, \tag{2.6}$$

linking the kinetic energy E_{kin} to the thermodynamic temperature T. The pressure P

combines kinetic and configurational (virial) contributions and is given by

$$P = \frac{NT}{V} + \frac{1}{dV}\sum_{i}^{N} \boldsymbol{r}_i \cdot \boldsymbol{F}_i. \tag{2.7}$$

In this thesis, a variety of systems is investigated using MD simulations, with details provided in the respective sections. All simulations are carried out with LAMMPS [36, 37].

2.1.2 Event-driven Brownian Dynamics Simulation

Brownian particle simulations of hard-disk systems allow to investigate slow relaxation processes in dense two-dimensional fluids. In such systems, particles i are characterized by their positions $\boldsymbol{r}_i(t)$, diameters σ_i, and translational diffusion coefficients $D_{\mathrm{t},i}$. The simulations are performed with periodic boundary conditions to resemble bulk-like behavior. Because relaxation times grow significantly near dynamical arrest, long simulations are usually required.

The dynamics are governed by a hybrid scheme, called event-driven Brownian dynamics (EDBD), which combines stochastic Monte-Carlo-like displacements with elastic collisions. The stochastic part of the dynamics follows the overdamped Langevin equation [38], discretized as

$$\Delta \boldsymbol{r}_i = \sqrt{2D_{\mathrm{t},i}}\, \boldsymbol{\xi}_i \Delta t, \tag{2.8}$$

where $\boldsymbol{\xi}_i$ denotes Gaussian random noise $\mathcal{N}(0,1)$ with unit variance. These displacements are translated into auxiliary pseudo-velocities,

$$\boldsymbol{v}_i = \frac{\Delta \boldsymbol{r}_i}{\Delta t} = \sqrt{\frac{2D_{\mathrm{t},i}}{\Delta t}}\, \boldsymbol{\xi}_i, \tag{2.9}$$

which are randomized independently at each Brownian timestep Δt. During a timestep, interactions are resolved using an event-driven scheme to handle collisions [39, 40].

For every particle pair (i, j), the algorithm predicts the collision time by solving

$$t_{i,j}^{\mathrm{coll}} = \frac{-\boldsymbol{r}_{i,j} \cdot \boldsymbol{v}_{i,j} - \sqrt{(\boldsymbol{r}_{i,j} \cdot \boldsymbol{v}_{i,j})^2 - \boldsymbol{v}_{i,j}^2 \left(\boldsymbol{r}_{i,j}^2 - (\sigma_i/2 + \sigma_j/2)^2\right)}}{\boldsymbol{v}_{i,j}^2}, \tag{2.10}$$

with $\boldsymbol{r}_{i,j} = \boldsymbol{r}_i - \boldsymbol{r}_j$ and $\boldsymbol{v}_{i,j} = \boldsymbol{v}_i - \boldsymbol{v}_j$, provided the particles are approaching, i.e., $\boldsymbol{r}_{i,j} \cdot \boldsymbol{v}_{i,j} < 0$. If a collision occurs within the interval Δt, the velocities are updated

according to

$$\boldsymbol{v}_i' = \boldsymbol{v}_i - \frac{2m_j}{m_i + m_j} \frac{\boldsymbol{r}_{i,j} \cdot (\boldsymbol{v}_{i,j} \cdot \boldsymbol{r}_{i,j})}{\boldsymbol{r}_{i,j}^2}, \tag{2.11}$$

$$\boldsymbol{v}_j' = \boldsymbol{v}_j + \frac{2m_i}{m_i + m_j} \frac{\boldsymbol{r}_{i,j} \cdot (\boldsymbol{v}_{i,j} \cdot \boldsymbol{r}_{i,j})}{\boldsymbol{r}_{i,j}^2}. \tag{2.12}$$

To maintain consistency with the fluctuation-dissipation theorem, effective particle masses are introduced [41],

$$m_i = \frac{\Delta t}{2\beta D_{\mathrm{t},i}}, \tag{2.13}$$

ensuring that the velocity distribution reproduces Maxwell–Boltzmann statistics with $\beta = \frac{1}{k_\mathrm{B} T}$. After all collisions within Δt are resolved, the velocities are randomized again, restarting the cycle.

The computational efficiency of this scheme relies on standard optimization techniques. Neighbor lists are used to restrict collision predictions to nearby particles, and priority queues are employed to manage ordering of events.

This approach can be extended to active Brownian particles (ABPs), where each particle gets an orientation φ and a self-propulsion velocity along it [42, 43]. The stochastic velocity update then becomes

$$\boldsymbol{v} = v_0 \begin{pmatrix} \cos\varphi \\ \sin\varphi \end{pmatrix} + \sqrt{\frac{2D_\mathrm{t}}{\Delta t}} \begin{pmatrix} \mathcal{N}(0,1) \\ \mathcal{N}(0,1) \end{pmatrix}, \tag{2.14}$$

while the orientation evolves as

$$\Delta\varphi = \nu_0 \Delta t + \sqrt{2D_\mathrm{r}}\mathcal{N}(0,1)\Delta t, \tag{2.15}$$

with v_0 denoting the self-propulsion speed, D_r the rotational diffusion coefficient, and ν_0 an intrinsic angular velocity.

A subtlety of Brownian dynamics arises from the character of pseudo-velocities. Since they are re-randomized at each timestep and serve only to propagate the system via collisions, they do not correspond to true microscopic dynamics with conserved kinetic energy. Between stochastic updates, the system behaves effectively like an NVE ensemble, with collisions conserving energy, but this invariance is broken by the Langevin updates. Overall, the algorithm provides an efficient scheme to realize Brownian motion of hard disks, capable of capturing the slow structural relaxation and collective behavior of dense colloidal systems.

A version of this algorithm was developed as part of my master's thesis and has been extensively tested for its reliability [42]. It is employed for the BD simulations of this thesis. Only passive Brownian particles are considered. Each system consists of 1000 particles, which has been found sufficient to suppress significant finite-size effects. A slight size-polydispersity is introduced to prevent crystallization and enable the investigation of systems approaching dynamical arrest. Initial configurations are generated by an algorithm similar to EDBD, in which particles are grown from points to their final sizes with the stochastic updates, while collisions with negative times remove small overlaps. These initial configurations at packing fraction ϕ are equilibrated through very long simulations, several times longer than the relaxation time. For each state point at a given ϕ, 200 independent simulations are performed for ensemble averaging, with additional ensembles obtained by assuming ergodicity and extracting sufficiently spaced sub-ensembles. Natural units with $D_{\mathrm{t,\,mean}} = 1.0$ and $\sigma_{\mathrm{mean}} = 1.0$ are employed.

2.1.3 System Properties

From trajectories $\{\boldsymbol{r}_i(t)\}$ obtained by simulations with either MD or BD, a wide range of structural and dynamical observables can be calculated [30, 44]. These quantities provide the link between microscopic particle motion and the description of matter given my statistical mechanics. They allow translating the raw positional information of particles into correlation functions and transport coefficients, which can be compared to theoretical predictions or experimental data.

Static Structure

An important measure of structural correlations in liquids and colloidal systems is the radial distribution function (RDF), also known as the pair correlation function. It measures the probability of finding a particle at separation $\boldsymbol{r}$ from a given reference particle. It is defined as

$$g(\boldsymbol{r}) = \frac{V}{N^2} \sum_{i=1}^{N} \sum_{i \neq j} \left\langle \delta(\boldsymbol{r} - \boldsymbol{r}_{i,j}) \right\rangle , \tag{2.16}$$

where $\boldsymbol{r}_{i,j}$ denotes the distance vector between particles i and j and $\langle ... \rangle$ the ensemble average [30].

The reciprocal-space analogue of the RDF is the (static) structure factor (SF),

$$S(\boldsymbol{q}) = \frac{1}{N} \sum_{i=1}^{N} \sum_{i \neq j} \left\langle \exp\left(-\mathrm{i}\boldsymbol{q} \cdot \boldsymbol{r}_{i,j}\right) \right\rangle , \tag{2.17}$$

which characterizes density correlations at wave vector $\boldsymbol{q}$. For isotropic systems, angular averaging reduces these functions to $g(r)$ and $S(q)$. They are obtained numerically from histograms of interparticle distances or wave vectors, averaged over sufficiently many simulation snapshots.

For multi-component systems, partial RDFs and partial SFs may be defined analogously by restricting the sums over particle indices to specific species pairs, such as A-A, B-B, or A-B correlations for a mixture of A and B particles. These allow a decomposition of the total structural correlations into their species-resolved contributions.

Dynamics

Dynamical behavior can be encoded in time-correlation functions, which track how fluctuations of microscopic variables evolve with time [30]. For a dynamical variable $\mathcal{A}$, a transport coefficient γ can be expressed in terms of the Green–Kubo relation,

$$\gamma = \int_0^\infty \langle \dot{\mathcal{A}}(t)\dot{\mathcal{A}}(0)\rangle \, \mathrm{d}t, \tag{2.18}$$

which relates macroscopic transport to the time-integral of microscopic current-current correlations. An equivalent form is provided by the Einstein relation,

$$\gamma = \lim_{t\to\infty} \frac{\mathrm{d}}{\mathrm{d}t}\frac{1}{2}\langle(\mathcal{A}(t) - \mathcal{A}(0))^2\rangle, \tag{2.19}$$

which connects transport coefficients to the long-time growth of the mean square differences of observables.

For d-dimensional particle positions, i.e., $\mathcal{A} = \boldsymbol{r}$, these relations yield the translational diffusion coefficient of particle i, either via the velocity auto-correlation function,

$$D_{\mathrm{t},i} = \frac{1}{d}\int_0^\infty \langle \boldsymbol{v}_i(t)\cdot\boldsymbol{v}_i(0)\rangle \, \mathrm{d}t, \tag{2.20}$$

or equivalently from the mean squared displacement (MSD),

$$D_{\mathrm{t},i} = \frac{1}{2d}\lim_{t\to\infty}\frac{\mathrm{d}}{\mathrm{d}t}\langle(\boldsymbol{r}_i(t) - \boldsymbol{r}_i(0))^2\rangle. \tag{2.21}$$

The MSD is given by

$$\langle \Delta\boldsymbol{r}^2\rangle(t) = \frac{1}{N}\sum_{i=1}^{N}\left\langle \left[\boldsymbol{r}_i(t+t_0) - \boldsymbol{r}_i(t_0)\right]^2\right\rangle, \tag{2.22}$$

and provides a direct measure of diffusion in the system.

Another dynamical observable is obtained by extending the SF into the time-domain, yielding the intermediate scattering function (ISF),

$$S(\boldsymbol{q}, t) = \frac{1}{N} \sum_{j=1}^{N} \sum_{k=1}^{N} \left\langle \exp\left(-\mathrm{i}\boldsymbol{q} \cdot [\boldsymbol{r}_j(t + t_0) - \boldsymbol{r}_k(t_0)]\right) \right\rangle , \tag{2.23}$$

which describes how density fluctuations at wave vector $\boldsymbol{q}$ decay over time. The ISF is commonly decomposed into a distinct part, describing correlations between different particles, and a self-part, which monitors the displacement of individual particles. The latter is the self-intermediate scattering function (SISF), defined as

$$S^{\mathrm{self}}(\boldsymbol{q}, t) = \frac{1}{N} \sum_{j=1}^{N} \left\langle \exp\left(-\mathrm{i}\boldsymbol{q} \cdot [\boldsymbol{r}_j(t + t_0) - \boldsymbol{r}_j(t_0)]\right) \right\rangle . \tag{2.24}$$

This function is directly linked to self-diffusion and represents one of the most widely used dynamical correlators in both simulations and experiments.

2.2 Neural Networks

Neural networks (NNs) are highly flexible function approximators that are increasingly employed for data-driven modeling tasks. They can learn complex, high-dimensional mappings from input data to target quantities without explicitly imposing functional forms. This makes them particularly suitable in cases where analytical models are either unavailable or too complex, but representative training data can be generated from simulations or experiments.

A neural network is organized as a sequence of layers, each consisting of multiple nodes, or neurons, that are interconnected by weighted connections. Information propagates through the network in a feed-forward manner. An input vector $\boldsymbol{x}$ is transformed linearly by weight matrices and bias vectors, followed by the application of a (non-linear) activation function f. The transformation performed at layer l can be written as

$$\boldsymbol{h}^l = f(\boldsymbol{w}^l \, \boldsymbol{h}^{l-1} + \boldsymbol{b}^l) , \tag{2.25}$$

with $\boldsymbol{x} = \boldsymbol{h}^0$ denoting the input, while $\boldsymbol{w}^l$ and $\boldsymbol{b}^l$ represent the trainable weights and biases of the layer. The output of the final layer L defines the network prediction, $\boldsymbol{y} = \boldsymbol{h}^L$. A schematic representation of such a network architecture is provided in fig. 2.1.

By chaining multiple layers, neural networks can represent highly non-linear functions with arbitrary accuracy and thus act as universal function approximators [6]. The number of layers, the number of nodes within each layer, and the choice of activation functions

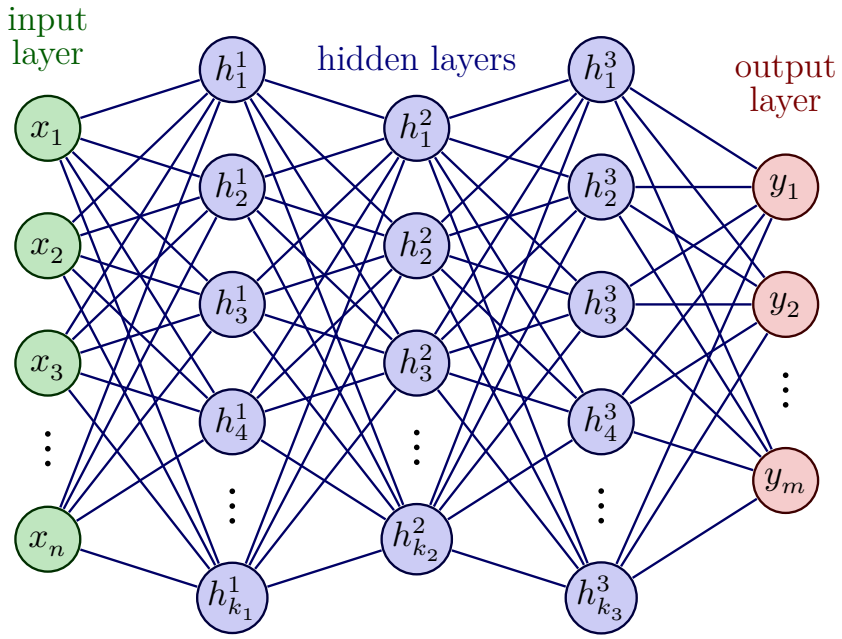


Figure 2.1: Illustration of the neural network architecture mapping an input vector $\boldsymbol{x}$ to an output vector $\boldsymbol{y}$ by transformation through multiple layers l. Activations f, weights $\boldsymbol{w}^l$ and biases $\boldsymbol{b}^l$ are applied between the layers according to $\boldsymbol{h}^l = f(\boldsymbol{w}^l \boldsymbol{h}^{l-1} + \boldsymbol{b}^l)$. Sketch adapted from [45].

constitute the important hyperparameters of the model and must be tuned to balance accuracy, efficiency, and generalization. Common activation functions include tanh, the rectified linear unit (ReLU), and the exponential linear unit (ELU).

The weights and biases are initialized randomly and are subsequently optimized during a training procedure. The optimization process aims to minimize a loss function $\mathcal{L}$, which quantifies the deviation between network predictions and reference data. Training is commonly performed using variants of stochastic gradient descent [46, 47], where gradients of the loss function with respect to the network parameters are computed via the backpropagation algorithm [48]. Efficient and accurate backpropagation relies on reverse-mode automatic differentiation [49], which applies the chain rule to the sequence of elementary operations executed during the forward pass. Because it differentiates the actual program rather than an analytic expression, it yields exact derivatives up to machine precision while maintaining computational cost comparable to a single forward evaluation.

A common challenge encountered during training is overfitting, in which the network learns noise or incidental patterns in the training set rather than the underlying relationships. In such cases, the training error continues to decrease, whereas the error on unseen data stagnates or even increases, indicating limited generalization. To mitigate overfitting and improve predictive accuracy on data not included in the training set, regularization

techniques such as early stopping [50], dropout [51], or weight decay [52] can be employed. By using a validation set separate of the training set, the extend of overfitting can be monitored.

The choice of network architecture, activation function, and training protocol strongly depends on the underlying problem, the available dataset, and the desired predictive accuracy. Another important aspect is the preprocessing and postprocessing of data, as both input and output often require suitable transformations, e.g., normalization, scaling, embedding, or reparameterization, to render them compatible with the architecture of the neural network. Training of neural networks is implemented in modern machine learning frameworks such as PyTorch [53] or TensorFlow [54], which provide efficient automatic differentiation, GPU acceleration, and optimized routines for these optimization problems.

2.3 Laplace Transform

The Laplace transform (LT), widely used in mathematics to solve integro-differential equations, converts functions of a real variable t to functions of a complex variable s [55]. The LT translates differentiation and integration in the time domain into multiplications and divisions in the Laplace domain.

Formally, the LT of a function $f(t)$ is defined as

$$\mathcal{L}\{f\}(s) = \widehat{f}(s) = \int_0^\infty \exp(-st)\, f(t)\, \mathrm{d}t, \tag{2.26}$$

where $s = \sigma + \mathrm{i}\omega$ is the complex frequency variable with real part σ and imaginary part ω. If the integral converges, the original function can be recovered through the inverse LT,

$$\mathcal{L}^{-1}\{\widehat{f}\}(s) = f(t) = \frac{1}{2\pi\mathrm{i}} \lim_{T\to\infty} \int_{\gamma-\mathrm{i}T}^{\gamma+\mathrm{i}T} \exp(st)\, \widehat{f}(s)\, \mathrm{d}s, \tag{2.27}$$

where $\gamma \in \mathbb{R}$ is chosen such that the integration contour lies within the region of convergence of $\widehat{f}(s)$.

A number of properties make the LT especially attractive in theoretical physics:

$$\mathcal{L}\{\partial_t f(t)\}(s) = s\mathcal{L}\{f\}(s) - f(0^-), \tag{2.28}$$

$$\mathcal{L}\left\{\int_0^t f(t')\, g(t-t')\, \mathrm{d}t'\right\}(s) = \mathcal{L}\{f\}(s) \cdot \mathcal{L}\{g\}(s), \tag{2.29}$$

$$\lim_{t\to 0^+} f(t) = \lim_{s\to\infty} s\, \mathcal{L}\{f\}(s), \tag{2.30}$$

$$\lim_{t\to\infty} f(t) = \lim_{s\to 0} s\, \mathcal{L}\{f\}(s). \tag{2.31}$$

These relations establish the correspondence between time-domain features and their Laplace-domain representations, such as short-time expansions and long-time asymptotics.

A particularly important special case occurs if $\hat{f}(s)$ converges on the imaginary axis, i.e., $\sigma = 0$. In this case, the LT reduces to the Fourier–Laplace transform (FT) [56]:

$$\hat{f}(\omega) = \mathcal{L}\{f(t)\}|_{s=\mathrm{i}\omega} = \int_0^\infty \exp(-\mathrm{i}\omega t)\, f(t)\, \mathrm{d}t, \tag{2.32}$$

with the corresponding inverse transform given by

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \exp(\mathrm{i}\omega t)\, \hat{f}(\omega)\, \mathrm{d}\omega. \tag{2.33}$$

For real-valued functions $f(t)$, the FT exhibits conjugate symmetry, i.e., even real part and odd imaginary part. Exploiting this property, the inverse relation can be simplified to

$$f(t) = \frac{1}{\pi} \mathrm{Re}\left(\int_0^\infty \exp(\mathrm{i}\omega t)\, \hat{f}(\omega)\, \mathrm{d}\omega \right). \tag{2.34}$$

In statistical physics, density correlation functions of liquids are real-valued and decay to zero at long times, ensuring that the FT exists as $\int_0^\infty |f(t)|\, \mathrm{d}t$ converges.

2.3.1 Laplace Transform of Exponentials

For exponential functions, the LT and FT can be evaluated analytically. A single decaying exponential,

$$f(t) = a \exp(-t/\tau), \tag{2.35}$$

yields

$$\hat{f}(s) = \frac{a}{s + 1/\tau}, \tag{2.36}$$

$$\hat{f}(\omega) = \frac{a\tau}{1 + \tau^2\omega^2} - \mathrm{i}\, \frac{a\tau^2\omega}{1 + \tau^2\omega^2}, \tag{2.37}$$

while its time derivative is given by

$$\frac{\partial f}{\partial t}(t) = -\frac{a}{\tau} \exp(-t/\tau), \tag{2.38}$$

with Laplace and Fourier–Laplace transform

$$\widehat{\frac{\partial f}{\partial t}}(s) = -\frac{a}{s\tau + 1}, \tag{2.39}$$

$$\widehat{\frac{\partial f}{\partial t}}(\omega) = -\frac{a}{1 + \tau^2\omega^2} + \mathrm{i}\,\frac{a\tau\omega}{1 + \tau^2\omega^2}. \tag{2.40}$$

These closed-form results are useful for numerical inversion schemes and testing stability of approximate algorithms.

2.3.2 Filon Method for Fourier Transforms

Laplace transforms of correlation functions are often computed numerically from data on discrete time grids. For oscillatory integrals, the Filon method provides an efficient numerical integration scheme by exploiting the known analytic integrals of exponential oscillations [57]. This method can be generalized to non-uniform, e.g., logarithmic, grids that are particularly well suited for functions spanning several orders of magnitude in time.

The Fourier–Laplace transform is then approximated as

$$\hat{f}(\omega) = \int_0^\infty \exp(-\mathrm{i}\omega t)\, f(t)\, \mathrm{d}t \approx \sum_{n=0}^{\infty} \delta t_n w_n \exp(-\mathrm{i}\omega t_n)\, f(t_n), \tag{2.41}$$

where δt_n denotes the step size and w_n are frequency-dependent weights. Assuming linear variation of $f(t)$ across each interval yields explicit expressions,

$$w_0 = \frac{1 - \mathrm{i}\omega\delta t_0 - \exp(-\mathrm{i}\omega\delta t_0)}{\omega^2 \delta t_0^2}, \tag{2.42}$$

$$w_n = \frac{\delta t_n/\delta t_{n-1}(1 - \exp(\mathrm{i}\omega\delta t_{n-1})) + 1 - \exp(-\mathrm{i}\omega\delta t_n)}{\omega^2 \delta t_n^2}, \quad n \neq 0. \tag{2.43}$$

Introducing a factor $\omega\delta t_n$ leads to

$$\hat{f}(\omega) = \sum_{n=0}^{\infty} \tilde{w}_n \frac{\exp(-\mathrm{i}\omega t_n)}{\omega}\, f(t_n), \tag{2.44}$$

with modified weights

$$\tilde{w}_0 = w_0 \cdot \omega\delta t_0 = -\mathrm{i} + \frac{1 - \exp(-\mathrm{i}\omega\delta t_0)}{\omega\delta t_0}, \tag{2.45}$$

$$\tilde{w}_n = w_n \cdot \omega\delta t_n = \frac{1 - \exp(\mathrm{i}\omega\delta t_{n-1})}{\omega\delta t_{n-1}} + \frac{1 - \exp(-\mathrm{i}\omega\delta t_n)}{\omega\delta t_n}. \tag{2.46}$$

For small $\omega\delta t_n$, these expressions can suffer from catastrophic cancellation. Taylor expan-

sions are therefore employed in this regime to improve numerical stability [58].

For inverse Fourier transform, the Filon method can be formulated analogously,

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \exp(\mathrm{i}\omega t)\, \hat{f}(\omega)\, \mathrm{d}\omega \approx \frac{1}{2\pi} \sum_{n=-\infty}^{\infty} \delta\omega_n w_n^{\mathrm{inv}} \exp(\mathrm{i}\omega_n t)\, \hat{f}(\omega_n), \tag{2.47}$$

and by exploiting conjugate symmetry for real-valued correlation functions, this can be written as

$$f(t) = \frac{1}{\pi} \sum_{n=0}^{\infty} \tilde{w}_n^{\mathrm{inv}} \frac{\exp(\mathrm{i}\omega_n t)}{t}\, \hat{f}(\omega_n), \tag{2.48}$$

with inverse weights

$$\tilde{w}_0^{\mathrm{inv}} = w_0^{\mathrm{inv}} \cdot t\delta\omega_0 = \mathrm{i} + \frac{1 - \exp(\mathrm{i}t\delta\omega_0)}{t\delta\omega_0}, \tag{2.49}$$

$$\tilde{w}_n^{\mathrm{inv}} = w_N^{\mathrm{inv}} \cdot t\delta\omega_N = \frac{1 - \exp(-\mathrm{i}t\delta\omega_{n-1})}{t\delta\omega_{n-1}} + \frac{1 - \exp(\mathrm{i}t\delta\omega_n)}{t\delta\omega_n}. \tag{2.50}$$

In practical applications, the Filon method for the Fourier transform is numerically stable up to intermediate frequencies, where oscillatory cancellations begin to dominate. The inverse transform, by contrast, is considerably more delicate. Since $\hat{f}(\omega)$ typically decays rather slowly, high-frequency contributions can dominate short-time behavior and amplify numerical noise. As a result, while the forward transformation tends to smooth this statistical noise in the input data, the inverse transformation reintroduces and even amplifies such fluctuations. This sensitivity is particularly problematic when the inverse transform is performed after a forward transform and additional operations.

3. Neural Network Potentials for Systems with Slow Dynamics

Computer simulations complement both theory and experiment in the study of condensed matter and molecular systems. Their success relies crucially on the accurate approximation of interatomic energies and forces, which govern the dynamics and thermodynamic properties of materials across scales. In theory, ab initio methods such as density functional theory (DFT) provide a systematic route to achieving this accuracy, being rooted in quantum mechanics. However, their computational cost renders them prohibitively expensive for large system sizes or for accessing the long time scales necessary to capture slow relaxation processes and collective phenomena [18].

Traditional empirical force fields, on the other hand, offer excellent computational efficiency but rely on fixed functional forms or parametrizations of specific model systems. While widely used, their transferability and accuracy are limited, often failing to capture structural and dynamical phenomena encountered in real-world materials [30, 59].

Machine learning potentials have emerged as a promising middle-ground, and neural network potentials (NNPs) in particular have become a widely adopted approach [8, 9, 11–15, 17]. The central idea of a NNP is to represent the total energy of a system as a sum of atomic contributions, which depend only on the local atomic environment of atoms. The environment needs to be encoded by descriptors, which are invariant under translation, rotation, and permutation of identical particles. A neural network then provides a non-linear mapping from descriptors to atomic energies, allowing for accurate and transferable approximations of potentials.

Forces follow directly as gradients of the learned energy with respect to atomic coordinates, such that molecular dynamics simulations can be performed with accuracy comparable to the reference data, yet at a fraction of the computational cost. In this sense, NNPs aim to combine the accuracy of first-principles methods with the efficiency of classical force fields, enabling studies of complex systems and long-time dynamics that are otherwise inaccessible. Indeed, NNPs have already demonstrated impressive performance across a broad range of physical systems [8–17].

It should be emphasized, however, that NNPs can at best reproduce the quality of their reference data. If the underlying model fails to capture essential physics, the trained potential will inherit these limitations. On the other hand, once trained, NNPs allow scaling to larger system sizes and longer time scales than what is feasible with the underlying model. This makes it possible to investigate emergent properties such as collective relaxation, cooperative dynamics, or structural transitions, which are often out of reach for DFT simulations.

A challenge lies in the fact that NNPs are commonly benchmarked against prediction errors on static configurations, or against structural observables of simple liquids and solids. Whether such metrics are sufficient indicators for stability and accuracy in systems with inherently slow dynamics, such as glass-forming liquids, is not obvious. Close to dynamical arrest, even small systematic deviations in force magnitudes or directions could accumulate, leading to qualitatively different long-time behavior.

In order to assess this, base models are required that are computationally efficient enough to allow direct comparison at long time scales. In this chapter, two classical systems are considered. The Lennard-Jones (LJ) potential [19–21] and the Voronoi potential [22, 23]. Binary mixtures of both are widely used to model glass-formers. These systems allow systematic tests of training strategies, energy and force reproduction, and performance of NNPs in reproducing slow dynamics.

While the simple LJ potential mostly serves for setting the baseline, the Voronoi potential introduces a complex energy landscape, where interactions are defined geometrically by (radical) Voronoi tessellations. Unlike LJ, the Voronoi interaction is inherently many-body, with forces depending non-trivially on the surrounding neighbors. This makes it ideal for testing the limits of NNPs, being structurally and dynamically complex, while still far less computationally expensive than electronic-structure methods.

Finally, as an excursion towards real materials, the case of Boron is considered based on AIMD reference data [60, 61]. Here, lessons learned from the model systems are transferred to improve training for a DFT system, demonstrating how insights from simple but well-controlled models can be leveraged for more realistic applications.

In the following sections, the different NNP architectures employed in this work are introduced, followed by systematic analyses of their training performance and their ability to reproduce structural and dynamical properties across the test systems.

3.1 Neural Network Potentials

The idea underlying neural network potentials (NNPs) is the decomposition of the total potential energy into atomic contributions, each depending only on the local environment of a given particle. This construction enables an accurate representation of the potential-energy surfaces (PES) with favorable scaling behavior with system size, since the complexity grows linearly with the number of atoms. The underlying physical assumption is that atoms predominantly interact with their nearby neighbors, such that distant contributions can be neglected beyond a finite cut-off radius.

The total energy E of a system of N atoms is expressed as a sum of atomic contributions

$$E = \sum_{i=1}^{N} E_i, \tag{3.1}$$

where atomic energies E_i depend only on the local neighborhood of atom i. This form ensures transferability across systems of different sizes, as the construction of atomic energies does not rely on global system properties. The mapping from local atomic environments to energy contributions is achieved through environment descriptors, which are provided as input to a feed-forward neural network (NN), introduced in sec. 2.2. The descriptors must encode the local environment in a way that preserves the fundamental invariances of the system. In particular, they must be invariant under translation, rotation, and permutation of identical atoms. Cartesian coordinates are thus unsuitable as direct input, since they change under these operations even though the energy remains invariant.

The local atomic environment of particle i is encoded into a set of M descriptors,

$$\boldsymbol{G}_i = \{G_{i,1}, G_{i,2}, \dots, G_{i,M}\}, \tag{3.2}$$

which are designed to uniquely characterize the neighborhood within the cut-off. The NN maps these descriptors onto an atomic energy $E_i(\boldsymbol{G}_i)$, such that the total potential energy can be expressed as

$$E = \sum_{i=1}^{N} E_i(\boldsymbol{G}_i). \tag{3.3}$$

Several different approaches to construct the descriptors $\boldsymbol{G}_i$ have been proposed. Common examples include atom-centered symmetry functions [11, 62] (sec. 3.1.1), the smooth overlap of atomic positions (SOAP) representation [13], environment matrices [15, 16] (sec. 3.1.2), and graph-neural network embeddings [17] (sec. 3.1.3). While the first two rely on explicit functional forms with hand-crafted parameters, the latter two utilize additional NNs to

embed and transform atomic coordinates into high-dimensional feature spaces. In all cases, descriptors must respect the invariances of the underlying physics while providing sufficient resolution to discriminate distinct environments.

Forces on atom k follow as the negative gradient of the total energy with respect to the atomic position $\boldsymbol{r}_k$,

$$\boldsymbol{F}_k = -\frac{\partial E}{\partial \boldsymbol{r}_k} = -\frac{\partial}{\partial \boldsymbol{r}_k} \sum_{i=1}^{N} E_i(\boldsymbol{G}_i) \tag{3.4}$$

$$= -\sum_{i=1}^{N} \sum_{m=1}^{M} \frac{\partial E_i}{\partial G_{i,m}} \frac{\partial G_{i,m}}{\partial \boldsymbol{r}_k}. \tag{3.5}$$

The first factor, $\partial E_i / \partial G_{i,m}$, is provided by the neural network and can be obtained efficiently via backpropagation due to the differentiable nature of the model. The second factor, $\partial G_{i,m} / \partial \boldsymbol{r}_k$, depends only on the definition of the descriptors and vanishes if atom k does not contribute to the descriptors of particle i. This locality drastically reduces the number of operations required to evaluate forces. A schematic of the NNP architecture is shown in fig. 3.1. Importantly, forces follow deterministically from the total energy.

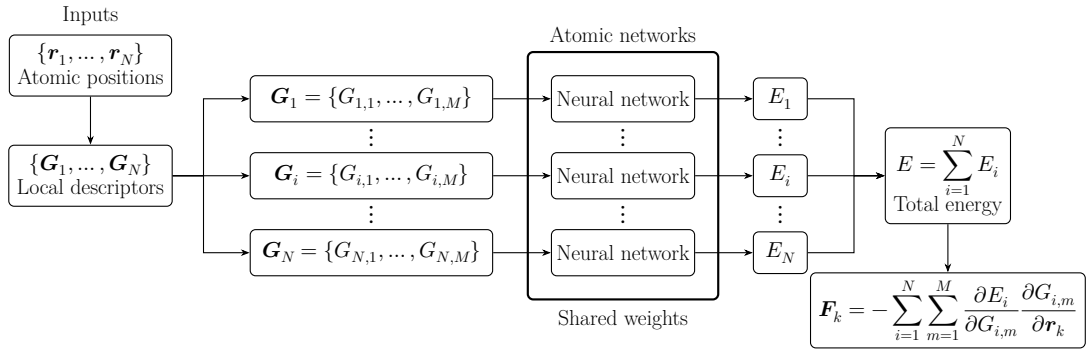


Figure 3.1: Schematic illustration of the neural network potential architecture. For each atom i, a set of local environment descriptors $\boldsymbol{G}_i$ is constructed from the positions of its N_i neighboring atoms, $\{\boldsymbol{r}_k\}_{N_i}$. These descriptors are passed through a shared feed-forward neural network to predict the atomic energy E_i. The total energy E is obtained as the sum over all atomic energies. Atomic forces $\boldsymbol{F}_k$ follow from the combination of backpropagation through the neural network and derivatives of the descriptors with respect to atomic coordinates.

The atomic energies E_i themselves do not necessarily carry a direct physical meaning, and during training they are often redistributed to minimize the global loss function. They should therefore be regarded as auxiliary variables whose only requirement is that their sum yields accurate predictions for total energies and forces. Reference data for training typically consist of atomic configurations with associated total energies and forces, obtained from first-principle simulations or high-quality empirical models. Once trained,

the neural network reproduces both static quantities (energies) and dynamic behavior (forces in molecular dynamics simulations). This makes NNPs highly suitable for large-scale simulations of condensed matter and materials systems [8, 9].

The accuracy of a neural network potential usually cannot be judged by direct inspection of the functional form, since the true potential-energy surface is unknown in complex systems. Instead, quality and transferability must be assessed indirectly by comparing simulation results obtained with the NNP to the reference model. Such comparisons typically involve both static observables, such as radial distribution functions, and dynamic observables, such as the self-intermediate scattering function. These validation steps are crucial to ensure that the learned potenial can be used reliably in simulations.

In the following, different methods to construct atomic descriptors will be described in detail.

3.1.1 High-Dimensional Neural Network Potentials

High-dimensional neural network potentials (HDNNP) were first introduced by Behler and Parrinello in 2007 [11, 62] as one of the earliest frameworks to approximate the potential-energy surface (PES) of atomic systems directly from sampled configurations [10]. The local environment of each atom is represented by a set of carefully designed descriptors, known as atom-centered symmetry functions, which are then used as input to a neural network.

The symmetry functions can encode both radial and angular information about the atomic neighborhood of atom i within a finite cut-off radius r_{cut}. For the radial part, a commonly used functional form is given by

$$G_i^{\mathrm{rad}} = \sum_{j \neq i} \exp\left[-\eta \left(r_{i,j} - r_s\right)^2\right] f_{\mathrm{c}}(r_{i,j}), \tag{3.6}$$

where $r_{i,j}$ denotes the distance between atoms i and j, and $f_{\mathrm{c}}(r)$ is a cut-off function that smoothly goes to zero at $r = r_{\mathrm{cut}}$. A typical choice is

$$f_{\mathrm{c}}(r) = \frac{1}{2}\left[\cos\left(\pi \frac{r}{r_{\mathrm{cut}}}\right) + 1\right]. \tag{3.7}$$

Angular information can be incorporated through three-body terms of the form

$$G_i^{\mathrm{ang}} = 2^{1-\zeta} \sum_{\substack{j,k \neq i \\ j<k}} (1 + \lambda \cos\theta_{i,j,k})^{\zeta} \exp\left[-\eta((r_{i,j} - r_s)^2 + (r_{i,k} - r_s)^2)\right] f_{\mathrm{c}}(r_{i,j}) f_{\mathrm{c}}(r_{i,k}), \tag{3.8}$$

where $\theta_{i,j,k}$ is the bond angle spanned by atoms $j\text{–}i\text{–}k$. The parameters η, ζ, λ, r_s, and r_{cut}

then define a specific descriptor. By combining mutliple functions with different parameter choices, a set of M descriptors $\boldsymbol{G}_i = \left\{G_{i,m}\right\}_{m\in\{1,\dots,M\}}$ is generated, characterizing the atomic environment of atom i. Examples of radial and angular symmetry functions for varying parameter values are illustrated in fig. 3.2.

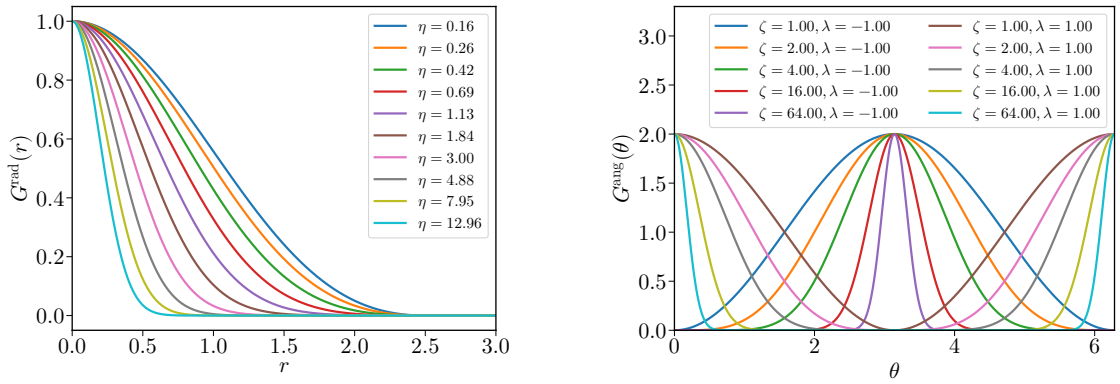


Figure 3.2: Examples of radial and angular symmetry functions, $G^{\mathrm{rad}}(r)$ and $G^{\mathrm{ang}}(\theta)$, shown as functions of the interparticle distance r and angle θ. For the angular descriptor, only the angle-dependent contributions are plotted. These descriptors serve as input to the HDNNP, with parameters generated according to the procedure proposed in [63].

The descriptor vector $\boldsymbol{G}_i$ is passed as input to a feed-forward neural network, which typically consists of two hidden layers and outputs a scalar atomic energy E_i. Differentiation of the network output with respect to atomic positions yields the forces acting on each atom. Because the symmetry functions are analytical, their derivatives can be computed efficiently, making this approach computationally inexpensive compared to descriptor schemes based on embedding networks.

For multi-component systems, separate sets of symmetry functions need to be defined for each pair (for G^{rad}) or triplet (for G^{ang}) combination of atomic species and each species is assigned its own neural network to predict atomic energies, which are then all summed to obtain the total energy of the system. This architecture ensures flexibility and accuracy but becomes increasingly expensive as the number of species grows, due to the combinatorial increase in required symmetry functions. Various extensions, such as weighted atom-centered symmetry functions [14], have been proposed to mitigate this limitation by incorporating species-dependent weightings directly into the descriptors. For systems with only two or three components, the computational overhead remains manageable because of the overall efficiency of the symmetry-function evaluation.

The HDNNP framework is implemented in the n2p2-package [12, 64], which provides both training routines and an interface to LAMMPS [65] for molecular dynamics simulations. The software evaluates the symmetry functions, propagates them through the neural network, and accumulates atomic contributions to predict the total potential energy of a

configuration. Training is performed by adjusting network weights based on a Kalman-filter optimization scheme, which exploits higher-order derivatives to obtain accurate gradients [12]. While the implementation itself is restricted to single-thread execution, substantial speed-ups can be achieved by parallelization over atomic environments and configurations, especially for sufficiently large systems where spatial decomposition becomes effective.

Despite their success, HDNNPs have a few notable limitations. The necessity of designing hand-crafted symmetry functions makes the setup of suitable descriptor sets a tedious task. Although empirical guidelines exist, e.g., [63], finding an optimal set typically requires substantial manual effort. Furthermore, the Gaussian-based functional forms lead to significant redundancy among descriptors, meaning that the number of required functions can often be reduced from an initially large set without loss of accuracy. The representation also encodes only radial two-body and angular three-body correlations. Nevertheless, this architecture has proven remarkably effective for a wide range of atomic and molecular systems.

3.1.2 Deep Potentials

To overcome limitations of hand-crafted descriptors, alternative network architectures have been developed in which the representation of the atomic environment is learned automatically through an embedding network, thereby removing the need for explicit functional forms.

One example of this approach is the Deep Potential Molecular Dynamics (DeePMD) framework [15, 16, 66, 67], which implements a descriptor constructed through neural network embeddings and provides its own efficient interface to LAMMPS. The default descriptor, denoted $\mathrm{se}_{\mathrm{e2,a}}$, incorporates both radial and angular information of atomic configurations while ensuring the invariances required. The local environment of each atom is restricted to atoms within a cut-off radius r_{cut}.

The novelty of these deep potentials lies in the definition of symmetry-preserving descriptors via trainable embedding networks. Translation invariance is ensured by constructing a generalized coordinate matrix for each central atom i,

$$(\tilde{R}^i)_j = \{s(r_{j,i}),\, x_{j,i}/r_{j,i},\, y_{j,i}/r_{j,i},\, z_{j,i}/r_{j,i}\}, \tag{3.9}$$

where j runs over all neighboring atoms within r_{cut} and $\boldsymbol{r}_{j,i} = \left(x_{j,i},\, y_{j,i},\, z_{j,i}\right)$ denotes the connecting vector between atoms j and i with distance $r_{j,i}$. The weighting function is defined as $s(r_{j,i}) = 1/r_{j,i}$, optionally multiplied by a smooth cut-off. Rotational invariance

is imposed by forming

$$\Omega^i = \tilde{R}^i(\tilde{R}^i)^{\mathrm{T}}, \tag{3.10}$$

while permutation invariance is incorporated through a learnable embedding matrix $D^i(s(r_{j,i}))$, generated by a neural network. The encoded feature matrix

$$G^i = (D^{i,1})^{\mathrm{T}}\,\Omega^i\,(D^{i,2}) \tag{3.11}$$

is thus invariant under translation, rotation, and permutation, and serves as the descriptor for atom i. Typically, $D^{i,2}$ is chosen as a submatrix of the full embedding matrix $D^{i,1} = D^i$, with the number of extracted columns determined by the so-called axis neuron.

The feature matrix G^i is flattened into a vector and passed to a fitting network, which maps it onto the atomic energy E_i. The procedure is analogous to the HDNNP construction but operates on learned descriptors rather than hand-crafted ones. Owing to the larger number of features, the fitting networks used for DeePMD are usually substantially larger than those in HDNNPs, which increases both representational power and computational cost.

Nevertheless, the approach offers several advantages. The hyperparameters are largely reduced to architectural choices such as the number of layers and neurons in the embedding and fitting networks, as well as the cut-off radius. This allows for stable convergence of the training procedure without extensive parameter searches. In addition, multiple species can be incorporated efficiently, with the coordinate matrix being partitioned into blocks for each species and one embedding network per species. This yields linear scaling of complexity, in contrast to the factorial growth of the HDNNP symmetry-function approach. For very large numbers of species, however, the method still becomes costly. To address this, the DeePMD framework has introduced concepts such as type embeddings, which reduce the number of required networks by encoding species information into a single embedding.

3.1.3 Neural Equivariant Interatomic Potentials

While HDNNPs and DeePMD are able to incorporate radial and angular information, their representations are ultimately restricted to relatively low-order correlations, and higher-order rotational features are not explicitly resolved. A more recent development in machine learning interatomic potentials is the use of graph-based neural networks, which aim to naturally encode local atomic environments via graphs to incorporate higher-order tensorial features.

A prominent example is the Neural Equivariant Interatomic Potential (NequIP) [17, 68–70], which employs an E(3)-equivariant graph-neural network [71] to construct atomic descriptors. An efficient implementation with a direct LAMMPS interface has also been made available [72].

Atoms are represented as nodes of a graph, and edges connect neighboring atoms within a cut-off radius r_{cut}. Each node is initialized with a scalar embedding of the atomic number Z_i, implemented via a trainable linear layer. These scalar node features are then transformed into tensorial features by a sequence of interaction blocks.

These features correspond to irreducible representations of the rotation group O(3), characterized by their rotational order $l = 0, 1, \ldots, L$ and parity $p = -1, +1$. Within each interaction block, features are propagated using equivariant convolutions. Resulting features of the same rotational order and parity are subsequently mixed by linear layers, while higher-order features above a cut-off l_{max} are discarded to control computational complexity. Convolutional filters are constructed from Clebsch-Gordan tensor products of the form

$$S_m^{(l)}(\boldsymbol{r}_{i,j}) = R(r_{i,j})\, Y_m^{(l)}(\hat{\boldsymbol{r}}_{i,j}), \tag{3.12}$$

where $\boldsymbol{r}_{i,j}$ is the vector between atoms i and j, $\hat{\boldsymbol{r}}_{i,j}$ is the corresponding unit vector, and $Y_m^{(l)}$ are spherical harmonics. The learnable parameters of the filter reside entirely in the radial function $R(r_{i,j})$, which is implemented as a feed-forward NN acting on radial basis embeddings. Non-linear activation functions and residual connections are applied at the end of each block.

A typical NequIP model consists of three to six interaction blocks. The scalar $l = 0$ features from the final block are concatenated and passed into an output network, which maps them onto atomic energies E_i. The total potential energy of the system is then obtained as a sum over all atomic contributions.

By construction, all operations in NequIP preserve translation, rotation, and permutation symmetries. The use of graph-neural networks allows atomic numbers to be directly embedded in node features, ensuring favorable scaling with the number of species. The different species are naturally incorporated without increasing model complexity.

Tensor-rank flexibility allows to systematically improve the accuracy of the network by increasing l_{max}, although this comes at significant computational cost. NequIP models contain a very large number of trainable parameters, and hyperparameter choices, such as the number of blocks, tensor orders, radial basis size, and embedding dimensions, which strongly affect both accuracy and efficiency. Training and inference are thus much more resource-intensive than for HDNNPs or DeePMD.

NequIP shows several decisive advantages, as it explicitly preserves all relevant symmetries, learns descriptors directly from atomic coordinates and species without hand-crafted forms, and leverages higher-order features that substantially increase descriptive power. Benchmark studies have shown that NequIP achieves superior accuracy and data efficiency compared to traditional neural network potentials, and exhibits excellent generalization capability even when trained on relatively small datasets [17].

3.1.4 Training Procedure

The training procedure for NNPs is largely independent of the specific network architecture and follows a general workflow based on learning atomic configurations. Modern network frameworks are optimized for this setting such that input-output pairs, combined with an appropriate choice of hyperparameters, are sufficient to train a model capable of reproducing the underlying PES.

The training pipeline begins with the generation of atomic configurations and their associated reference data. Such data is typically obtained by sampling the phase-space of the system through MD simulations at selected densities, temperatures, and, where applicable, species compositions. Sampling from physically relevant regions of phase-space is particularly advantageous, since broad coverage of local atomic environments generally improves the transferability and robustness of the resulting model. Although any configuration could be used for training, networks trained on physically important states tend to perform more reliably in simulations.

Reference systems at the desired density are first equilibrated in the canonical (NVT) ensemble and subsequently sampled in either the NVT or NVE ensemble. It is often beneficial to supplement the equilibrium data with out-of-equilibrium configurations, e.g., those generated during rapid quenching or melting, or those encountered during equilibration. Including such structures can stabilize the NNP in simulations, which may explore out-of-equilibrium regions of phase-space. Each configuration contains atomic positions, atomic forces, the total potential energy, and, for mixtures, the corresponding atom types.

To ensure that the network is learning a generalizable mapping, the sampled data is partitioned into three subsets:

- a training set, used for optimization of the network weights,
- a validation set, used to monitor and evaluate network performance during training, and,
- an independent test set, used exclusively for the final assessment of model accuracy.

With a dataset defined, the network is trained using stochastic optimization. Training proceeds in batches of configurations, and one epoch is defined as a complete pass through the training set. The loss function measures the error between network predictions and reference data. For NNPs, the loss is typically defined as a combination of the total energy error $\mathcal{L}_{\mathrm{E}}$ and the force error $\mathcal{L}_{\mathrm{F}}$,

$$\mathcal{L} = \mathcal{L}_{\mathrm{E}} + \mathcal{W}_{\mathrm{F}}\,\mathcal{L}_{\mathrm{F}}, \tag{3.13}$$

where $\mathcal{W}_{\mathrm{F}}$ is a relative weight balancing the importance and magnitude of energy and force contributions. Often, the mean squared error (MSE) is employed as the error metric, although other choices are possible. The parameter $\mathcal{W}_{\mathrm{F}}$ can either be fixed or changed dynamically during training. Although training with energies alone is possible, including forces in the loss is strongly preferred, as forces provide a much richer set of constraints on the PES and significantly improve accuracy.

During each training iteration, the network parameters are updated according to an optimization strategy such as mini-batch gradient descent or one of its adaptive variants, controlled by a learning rate l_{r} determining the step size in parameter space. A common strategy is to gradually decrease the learning rate over the course of training in order to achieve convergence. Multiple epochs are required for convergence, and the order of batches within each epoch is randomized to improve generalization and reduce overfitting.

After successful training, the performance of the network is evaluated using the test set, ensuring that the final error metrics are independent of the training data. If the network achieves satisfactory accuracy, it can be used in MD simulations substituting the reference potential. Validation of the trained potential is then performed at the level of physical observables by comparing both structural quantities, such as radial distribution functions, and dynamical quantities, such as MSDs or SISFs, against the reference model.

3.2 Lennard-Jones Potential

Neural network potentials are commonly employed to approximate the potential-energy surfaces (PES) of complex systems. However, the development and validation of NN models can be studied by beginning with simple benchmark systems. Classical model potentials are defined analytically and are computationally inexpensive, which allows systematically testing NNPs and developing good practices for training and validation. In contrast to ab initio data, where reference values may already contain numerical noise and approximations, classical model potentials provide exact reference energies and forces for every configuration, thereby enabling an unambiguous assessment of accuracy.

A particularly suitable choice for such a benchmark is the Lennard-Jones (LJ) potential. Introduced as a general two-body model with a repulsive r^{-n} and an attractive r^{-m} contribution to improve viscosity models of gases [20, 73–76], it has become one of the most widely used model interactions in statistical mechanics and molecular simulation [19]. In molecular dynamics, the standard (12, 6)-form of the LJ potential is typically employed, balancing a steep short-range repulsion with a long-range attractive tail.

The LJ potential is versatile. It can describe systems ranging from dimers to bulk phases, from low to high densities, and from equilibrium liquids to supercooled states. Since the physics of LJ systems are extremely well studied, deviations between reference simulations and neural network predictions can be attributed directly to shortcomings of the network, rather than to poorly understood physical effects.

Beyond a one-component system, mixtures offer an additional layer of complexity without greatly increasing the difficulty of the underlying interactions. In particular, the binary Kob–Andersen (KA) mixture [21] has become a de-facto standard model for glass-forming liquids. Designed to avoid crystallization, the KA system exhibits slow relaxation and characteristic of glassy dynamics [77], making it an excellent benchmark for testing whether NNPs can reproduce subtle dynamical features.

The simplicity and flexibility of the LJ and KA models make them ideal candidates for assessing the performance of NNP. In this section, NNPs are trained on LJ- and KA-configurations and tested for their ability to reproduce the underlying systems. Deviations between neural network predictions and reference simulations, whether arising from deliberate use of poor training practices or from intrinsic limitations, are analyzed to highlight the limits and potential pitfalls of neural network potential development.

Insights gained are not limited to a specific descriptor architecture. Rather, they provide general guidance that is transferable across neural network models. In this sense, the LJ and KA systems provide a foundation for developing, testing, and refining neural network potentials in a controlled and computationally accessible setting.

3.2.1 Network Architecture

High-dimensional neural network potentials (HDNNPs), introduced in sec. 3.1.1, are employed for the Lennard-Jones interaction potential. All networks are implemented and trained using the n2p2-package.

Given the pairwise and radially symmetric nature of the LJ potential, atom-centered radial symmetry functions constitute a natural choice of descriptor. Angular symmetry functions could also be used, since they contain a radial contribution. However, when radial symmetry functions already cover the relevant range of distances, the additional angular terms carry negligible weight in the final NNP and do not improve training performance. The cut-off radius r_{cut} for the descriptors is set equal to the truncation distance of the LJ-interaction.

The parameters for the radial symmetry functions are generated according to the algorithm proposed in [63], with $r_s = 0$ and a total of $N_{\text{SF}} = 10$ values of η distributed uniformly between 0 and r_{cut}. This construction ensures that the full radial range is covered without gaps, and that redundant overlap between functions is minimized. This number of symmetry functions per species interaction was found sufficient in all cases studied.

The neural network consists of two hidden layers with ten neurons each and hyperbolic tangent activation functions are used between layers. For a system with N_{spec} species, the architecture can be summarized as

$$N_{\text{spec}} \times N_{\text{SF}} \times 10 \times 10 \times 1. \tag{3.14}$$

This modest network size is fully sufficient to capture the Lennard-Jones potential.

Training is performed by minimizing a combination of energy and force errors. Optimization is carried out using the Kalman-filter-based gradient descent algorithm, which has been shown to outperform conventional optimizers such as stochastic gradient descent or ADAMW in this framework [12]. In particular, convergence using ADAMW was observed to be both slow and unstable in this setting, possibly due to implementation details. The training procedure is straightforward once a sufficiently diverse dataset is provided, and all investigated LJ systems exhibit qualitatively similar convergence behavior. An example of a training history is shown in fig. 3.3.

Typically, the energy loss converges rapidly, often within the first few epoch, while the force error requires more iterations to reach stability. This reflects the higher sensitivity of the force landscape compared to the total energy. The noise observed in the energy loss curve is thus a byproduct of the already well-converged energy error, where the force convergence dominates the training behavior. Ultimately, the LJ potential is reliably learned by the

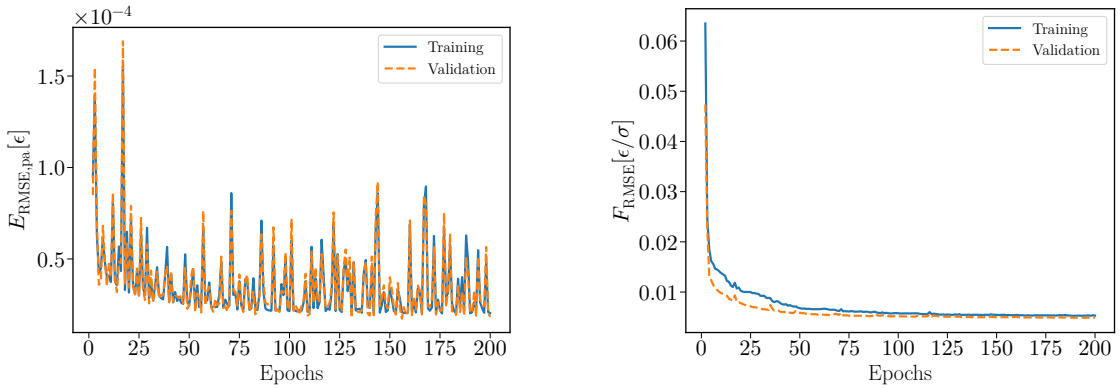


Figure 3.3: Example learning curves for energies (left) and forces (right) during training of the HDNNP on Lennard-Jones configurations. While the energy error converges rapidly, the force error requires substantially more training iterations due to the higher complexity of the force landscape. Training errors are shown as running averages within each epoch.

HDNNP, even for relatively small networks. Although even smaller architectures, with as few as five symmetry functions, could reproduce the potential, such aggressive reduction requires additional tuning and optimization that is unnecessary for a proof-of-principle.

3.2.2 Single-Component Lennard-Jones System

First, the simple single-component $(12, 6)$ Lennard-Jones system is considered. The potential between two particles separated by a distance r is defined as [30]

$$V_{\mathrm{LJ}}(r) = 4\epsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right], \tag{3.15}$$

where ϵ sets the energy-scale and σ defines the length-scale. The energy is pair-wise additive and the total energy is obtained as a sum over all particle pairs. In the following, reduced units are employed with $\epsilon = 1.0$ and $\sigma = 1.0$.

For efficient simulations, the potential is truncated at a finite cut-off distance r_{cut} in order to reduce computational cost [30, 78, 79]. Since this truncation introduces discontinuities, different cut-off schemes have been proposed. Two such variants are investigated, as they influence not only equilibrium properties such as the location of the critical point, but also the nature of the training data.

The first scheme involves shifting the energy to zero at the cut-off,

$$V_{\mathrm{LJ,ES}}(r) = \begin{cases} V_{\mathrm{LJ}}(r) - V_{\mathrm{LJ}}(r_{\mathrm{cut}}), & r \leq r_{\mathrm{cut}}, \\ 0, & r > r_{\mathrm{cut}}, \end{cases} \tag{3.16}$$

known as the energy-shifted potential [78]. While this modification ensures continuity of the energy, the corresponding forces are discontinuous at r_{cut}. As a result, the total energy is not conserved in microcanonical simulations. For neural network training, this provides an interesting test case, since the descriptors of neural network potentials naturally enforce continuity with respect to atomic coordinates.

An additional correction to the potential fixes the discontinuity such that both the energy and the forces vanish smoothly at the cut-off,

$$V_{\text{LJ,FS}}(r) = \begin{cases} V_{\text{LJ}}(r) - V_{\text{LJ}}(r_{\text{cut}}) - (r - r_{\text{cut}}) \left.\frac{\partial V_{\text{LJ}}(r)}{\partial r}\right|_{r=r_{\text{cut}}}, & r \leq r_{\text{cut}}, \\ 0, & r > r_{\text{cut}}, \end{cases} \tag{3.17}$$

referred to as the force-shifted potential [79]. Given that NNPs yield continuous forces through their differentiable structure, improved performance is expected when training on the force-shifted reference data compared to the energy-shifted variant.

The effect of dataset composition is also of interest for this model system. Training exclusively on configurations sampled from dilute states essentially teaches the network the bare two-body interaction, but leads to poor generalization and unstable extrapolation to dense states. In contrast, datasets including dense configurations provide the network with the diversity of local environments necessary to reproduce the full phase behavior.

Network Training

The training procedure follows the general workflow described in sec. 3.2.1. To disentangle network performance from potential dataset artifacts, an extensive set of configurations was constructed. A total of 50 000 configurations with 256 atoms each were generated, ensuring that the results are not biased by dataset size or sampling issues. The number of atoms is chosen to match the typical system size of a DFT calculation. Although preliminary tests indicated that far fewer samples already yield satisfactory NNPs, the larger dataset guarantees robustness and is therefore preferred here.

Two distinct datasets are prepared for both methods of shifting the potential. One is restricted to dense states with number densities $\rho \in [0.8, 1.25]$, and the other covers a wider range of $\rho \in [0.3, 1.25]$. The dense dataset contains only dense liquid and solid configurations, whereas the extended dataset spans most of the Lennard-Jones phase-diagram. For both datasets, configurations are sampled in the microcanonical (NVE) ensemble, with temperatures ranging from $T = 0.01$ to $T = 2.0$.

This setup allows for a investigation of the effect of dataset composition. Training exclusively on dense configurations leads to the absence of isolated particle-pairs, such that the bare

pair-potential is not explicitly sampled. In principle, the network can still recover the correct interaction through the pair-wise additive nature of the Lennard-Jones potential, provided that the non-linear activation functions operate in a quasi-linear regime, i.e., the tanh activations receive small arguments. In contrast, including dilute configurations explicitly presents the network with isolated particle pairs and directly encodes the pair potential into the training process. While dilute configurations alone would suffice to learn the two-body interaction, including dense states reduces extrapolation artifacts by exposing the network to a broader range of local environments.

For both datasets, separate models were trained on reference data generated from the energy-shifted and force-shifted variants of the Lennard-Jones potential. Both energies and forces are included for the optimization, and no special hyperparameter tuning was required to obtain satisfactory convergence.

The resulting testset errors are reported in tab. 3.1 for the energy-shifted potential and in tab. 3.2 for the force-shifted potential. While energy errors are very small across all models, they cannot be compared directly between datasets due to differences in the magnitude of reference energies and forces. As expected, including the force contribution in the loss significantly improves the accuracy of force predictions, albeit at the cost of increased energy errors. Since forces determine the dynamics in simulations, this trade-off is acceptable, particularly given that the energy errors remain negligible.

Overall, the training for the Lennard-Jones system is straightforward. All networks converge reliably, and the resulting NNPs achieve excellent accuracy with respect to both energies and forces.

Dataset	$\mathcal{W}_\mathrm{F}$	$E_\mathrm{RMSE,pa}$	F_RMSE
Dense	0	$1.922 \cdot 10^{-4}$	$6.538 \cdot 10^{-2}$
Dense	1	$1.977 \cdot 10^{-4}$	$4.882 \cdot 10^{-2}$
Dense	10	$3.277 \cdot 10^{-4}$	$3.946 \cdot 10^{-2}$
Full	0	$2.476 \cdot 10^{-4}$	$7.075 \cdot 10^{-2}$
Full	1	$3.003 \cdot 10^{-4}$	$5.452 \cdot 10^{-2}$
Full	10	$4.206 \cdot 10^{-4}$	$4.232 \cdot 10^{-2}$

Table 3.1: Per-atom total energy and force errors for neural network potentials of the energy-shifted Lennard-Jones potential, trained with different datasets and force-loss weights $\mathcal{W}_F$.

Dataset	$\mathcal{W}_\mathrm{F}$	$E_\mathrm{RMSE,pa}$	F_RMSE
Dense	0	$1.906 \cdot 10^{-5}$	$6.322 \cdot 10^{-3}$
Dense	1	$2.032 \cdot 10^{-5}$	$4.868 \cdot 10^{-3}$
Dense	10	$4.590 \cdot 10^{-5}$	$4.355 \cdot 10^{-3}$
Full	0	$2.491 \cdot 10^{-5}$	$7.374 \cdot 10^{-3}$
Full	1	$2.568 \cdot 10^{-5}$	$5.991 \cdot 10^{-3}$
Full	10	$2.901 \cdot 10^{-5}$	$4.409 \cdot 10^{-3}$

Table 3.2: Per-atom total energy and force errors for neural network potentials of the force-shifted Lennard-Jones potential, trained with different datasets and force-loss weights $\mathcal{W}_F$.

Potential Evaluation

The training of neural network potentials is guided by minimizing prediction errors with respect to energies and forces. However, small numerical errors in these metrics do not necessarily guarantee that the resulting potential will reproduce the correct physics in MD simulations. The quality of a trained potential should therefore also be assessed by independent means, most notably through the comparison of structural and dynamical observables obtained in simulations. For the LJ system, a more direct evaluation is possible. Owing to its analytical pair-wise additive form, it can be examined how well the underlying pair potential is reproduced, as illustrated in fig. 3.4. This type of analysis is unique to simple model systems and cannot be performed for complex many-body interactions or ab initio reference data, where the true functional form is unknown.

For the energy-shifted LJ potential, NNPs trained exclusively on dense configurations exhibit systematic deviations. Specifically, the learned pair potentials display a shift in the energy scale, and the corresponding pair forces are poorly reproduced due to extrapolation errors when the network encounters an isolated two-body environment. When the full range of densities is provided, i.e., dilute states are included in the training data, the pair potential is recovered. A small shift in energy is observed when forces are included in the loss function, which likely arises from compensation effects of balancing energy and force contributions in the loss function.

An interesting feature of NNPs is seen for the behavior at the cut-off. While the reference energy-shifted potential shows a discontinuity of the forces at r_cut, the NNP yields smooth behavior due to the cut-off functions in the descriptors. As a result, the force discontinuity, which is undesirable for molecular dynamics simulations anyways, is not represented. Again, potentials trained on the full dataset outperform those trained solely on dense states, and models including forces slightly improve the representation of forces near the cut-off compared to energy-only training.

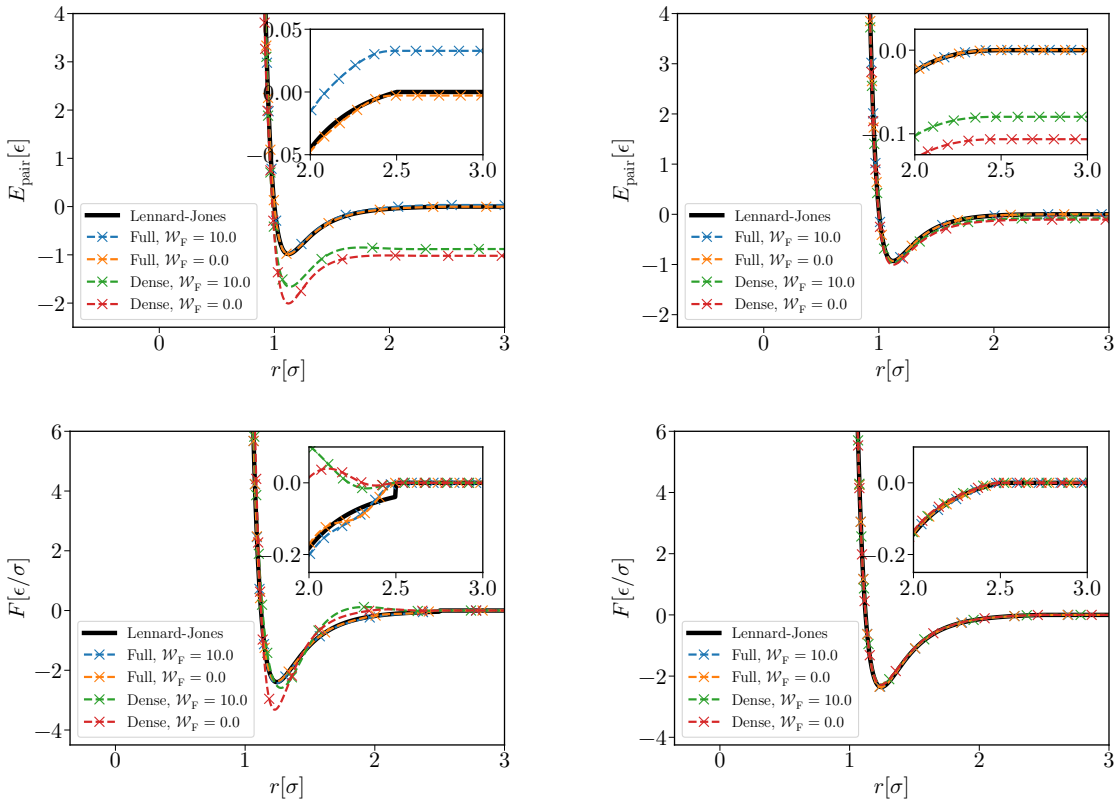


Figure 3.4: Comparison of pair potentials between the energy-shifted (left) and force-shifted (right) Lennard-Jones potentials (solid) and the corresponding neural network potentials (dashed), trained on only dense or dilute and dense configurations with different force-loss weights $\mathcal{W}_{\mathrm{F}}$.

For the force-shifted Lennard-Jones potential, the situation is different. Here, both the energy and forces are continuous at the cut-off by construction, and the NNPs reproduce these properties accurately. Pair forces are captured almost perfectly, regardless of whether the training set is restricted to dense states or includes also dilute configurations, and the accuracy appears largely independent of the relative force weight $\mathcal{W}_{\mathrm{F}}$. Pair energies are also well reproduced, although dense-only training sets produce a small shift relative to the reference. Including forces in the training loss tends to reduce this shift, though the effect may also fall within the statistical variance of the training procedure.

Since the networks are trained to represent the total energy of an N-body system, reproducing the effective two-body interactions is not guaranteed. In particular, the treatment of non-smooth forces at the cut-off is an issue, and while neural network potentials smooth out such discontinuities, even small inconsistencies could affect structural and dynamical observables in subtle ways.

A direct way to assess the suitability of a given NNP is to test its ability to conserve energy. In microcanonical (NVE) simulations, the only source of energy drift should be the discretization error introduced by the integration scheme. For Velocity-Verlet integration, this drift is expected to scale quadratically with the timestep, i.e., $\Delta E \propto \Delta t^2$ [80], leading to small, bounded fluctuations of the total energy when the underlying forces are continuous. However, if the forces are not continuous, this scaling breaks down and the energy error can exhibit systematic drifts even for small timesteps.

To test this, short simulations are carried out in the NVE ensemble, starting from identical initial configurations but with varying integration timesteps Δt, while keeping the total simulation time fixed. The deviation of the conserved total energy is then monitored and compared between the LJ potential and the NNPs trained on the full dataset.

The results for the energy-shifted potential are shown in fig. 3.5. Due to the discontinuity of the forces at the cut-off, the reference potential fails to exhibit the expected Δt^2 scaling of the discretization error and instead produces an irregular energy drift. In contrast, the neural network potential, which enforces smooth forces by construction, effectively "repairs" this deficiency and recovers the correct Δt^2 scaling behavior.

The situation is different for the force-shifted potential, where the reference potential is continuous in both energies and forces. As shown in fig. 3.6, the NNP reproduces the underlying behavior nearly perfectly for both dilute and dense training sets. In this case, the discretization error of the total energy agrees quantitatively with that of the reference model.

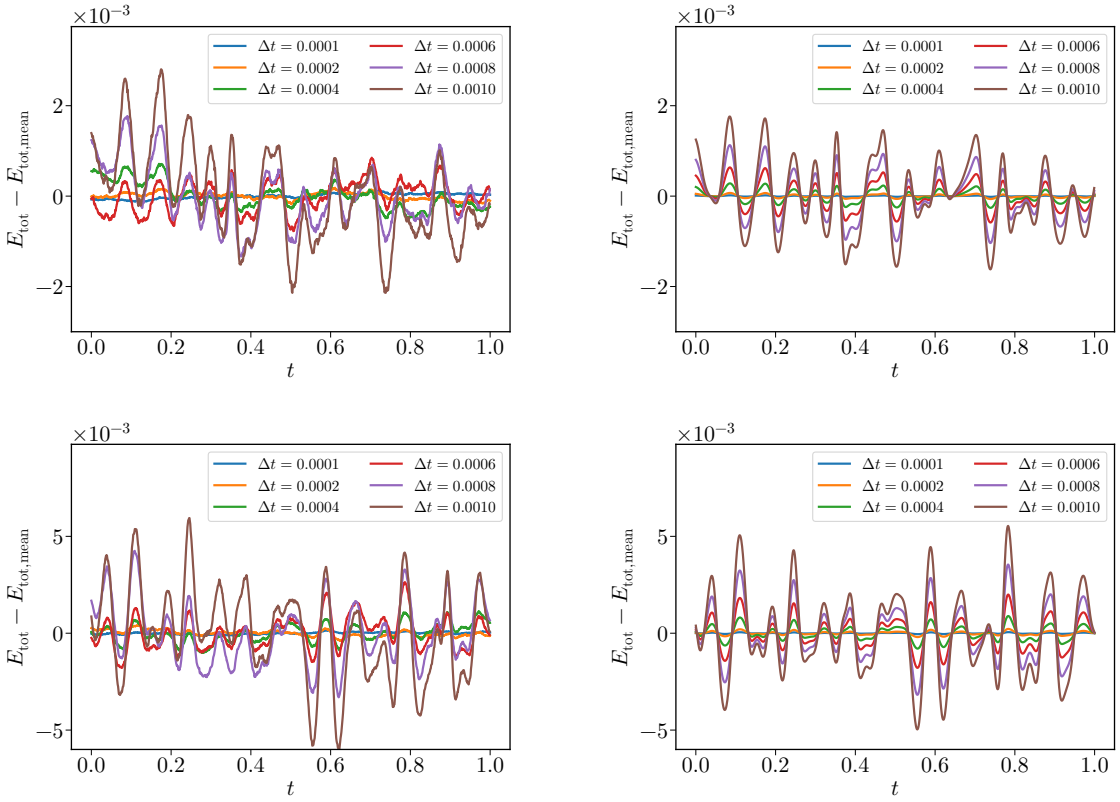


Figure 3.5: Comparison of microcanonical total energy fluctuations for the energy-shifted Lennard-Jones potential (left) and the corresponding neural network potential (right), trained on the full dataset with force weight $\mathcal{W}_{\mathrm{F}} = 10.0$. Results are shown for $T = 1.0$ and densities $\rho = 0.3$ (top) and $\rho = 0.8$ (bottom) at different timesteps Δt.

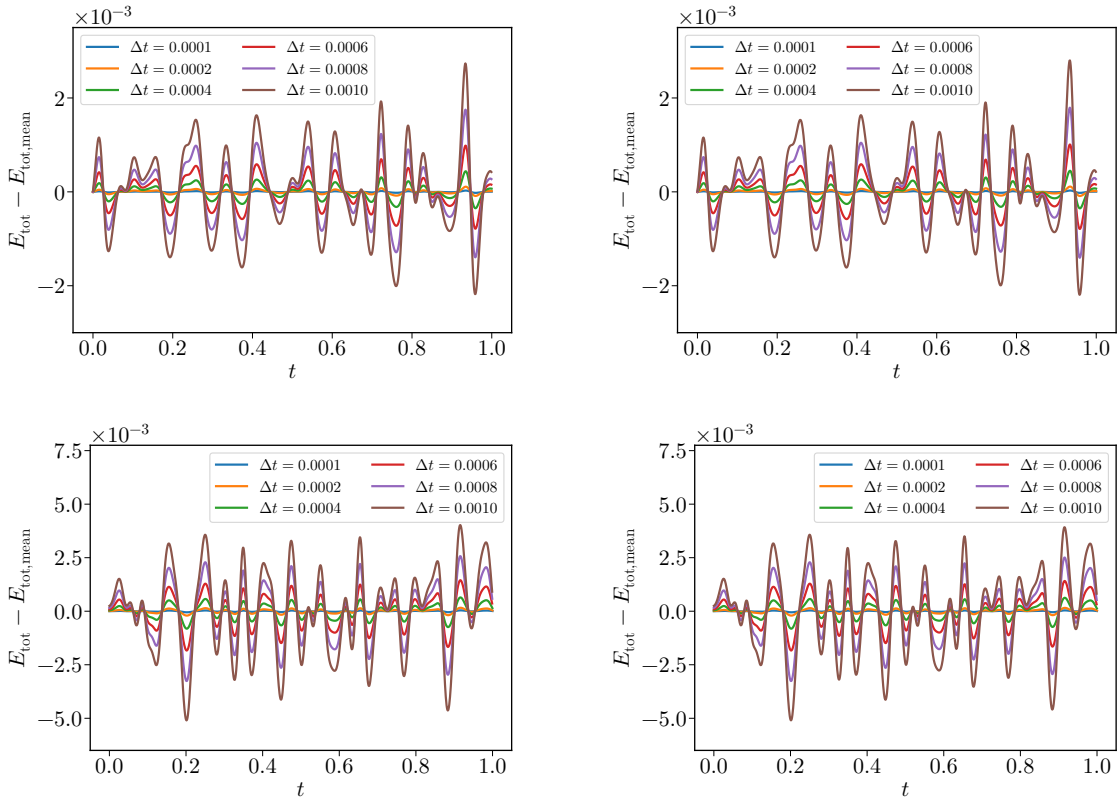


Figure 3.6: Comparison of microcanonical total energy fluctuations for the rorce-shifted Lennard-Jones potential (left) and the corresponding neural network potential (right), trained on the full dataset with force weight $\mathcal{W}_{\mathrm{F}} = 10.0$. Results are shown for $T = 1.0$ and densities $\rho = 0.3$ (top) and $\rho = 0.8$ (bottom) at different timesteps Δt

A more quantitative view of the discretization behavior is obtained from the standard deviation of the total energy discretization error, shown in fig. 3.7 for both cut-off schemes and their neural network counterparts. For the force-shifted LJ potential, as well as for all trained NNPs, the expected Δt^2 scaling is found. By contrast, the energy-shifted potential itself deviates from this behavior due to the discontinuity of the forces at the cut-off. Despite being trained on reference data from a discontinuous model, the neural network potentials effectively regularize the physics of the underlying system.

These results demonstrate that NNPs possess a certain corrective capacity, where they smooth out unphysical features such as force discontinuities, thereby producing models that are in some respects superior to their reference, but possibly altering fine details required for reproducibility and transferability.

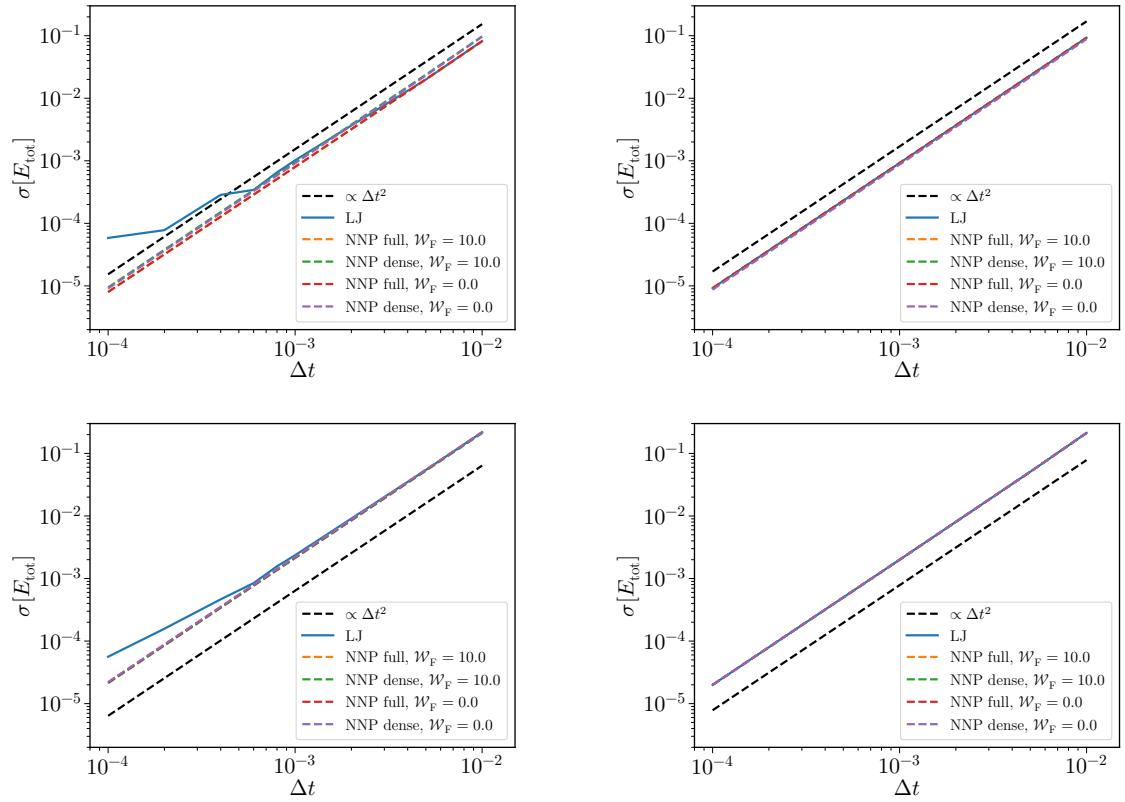


Figure 3.7: Standard deviation of microcanonical total energy fluctuations as a function of timestep Δt for the energy-shifted (left) and force-shifted (right) Lennard-Jones potentials, compared to neural network potentials (dashed) trained on different datasets and with varying force weights $\mathcal{W}_{\mathrm{F}}$. Results are shown for $T = 1.0$ at densities $\rho = 0.3$ (top) and $\rho = 0.8$ (bottom).

Simulation Comparison

Beyond reproducing pair potentials and verifying energy conservation, the quality of NNPs should also be assessed in terms of their ability to reproduce structural and dynamical properties of the reference system. To this end, MD simulations are performed and compared between the LJ potential and trained NNPs. The analysis is limited to the force-shifted LJ potential trained with force contributions in the loss function. The qualitative results for the energy-shifted variant are essentially identical once the cut-off-related differences discussed previously are accounted for.

All simulations are carried out in the microcanonical (NVE) ensemble with $N = 864$ particles over $5 \cdot 10^6$ timesteps with time-step of $\Delta t = 10^{-3}$. Random starting configurations are obtained from equilibrated NVT simulations. A standard LJ simulation requires approximately $60 - 90$ min on a single core of an AMD EPYC 7702 processor, whereas the corresponding NNP simulations using n2p2 with ten symmetry functions require $70 - 100$ h under identical conditions. Thus, the neural network approach is slower by a factor of roughly 70, reflecting the number of operations required for descriptor evaluation and network calculations at each timestep. They scale identically with system size.

Comparisons were performed for a variety of fluid and crystalline states. To obtain statistically robust results, four independent ensembles are simulated at each state point, and additional averaging over sub-ensembles is performed where applicable. Example comparisons are shown in fig. 3.8 for $\rho = 0.75$ and in fig. 3.9 for $\rho = 1.0$ at various temperatures T. In all cases, the structural and dynamical observables obtained with the NNPs show almost perfect agreement with those of the reference LJ system. Deviations remain within the statistical fluctuations expected for finite-size ensembles. This is particularly noteworthy, since for the network trained only on dense configurations, simulations at $\rho = 0.75$ lie outside the phase-space region used for training.

This level of agreement is observed consistently across a wide range of densities and temperatures beyond those explicitly shown here. The results demonstrate that the trained neural network potentials are of high quality, capable of reproducing both the static and dynamic behavior of the underlying model system. It should be emphasized, however, that this excellent performance is facilitated by the simplicity of the LJ potential.

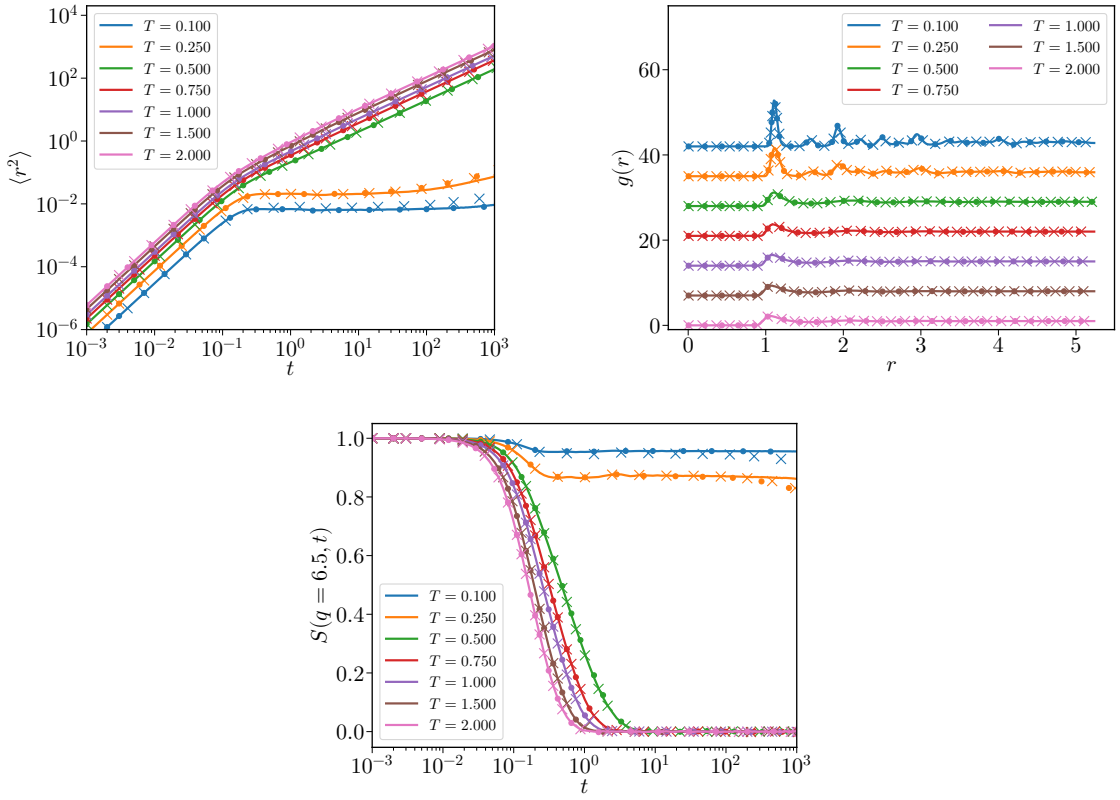


Figure 3.8: Comparison of MSDs (top left), RDFs (top right), and SISFs (bottom) between the force-shifted Lennard-Jones potential (solid lines) and the corresponding neural network potentials trained on the full dataset (crosses) and on dense configurations only (dots), with force weight $\mathcal{W}_\mathrm{F} = 10.0$, at density $\rho = 0.75$ and different temperatures T. RDFs are shifted for clarity.

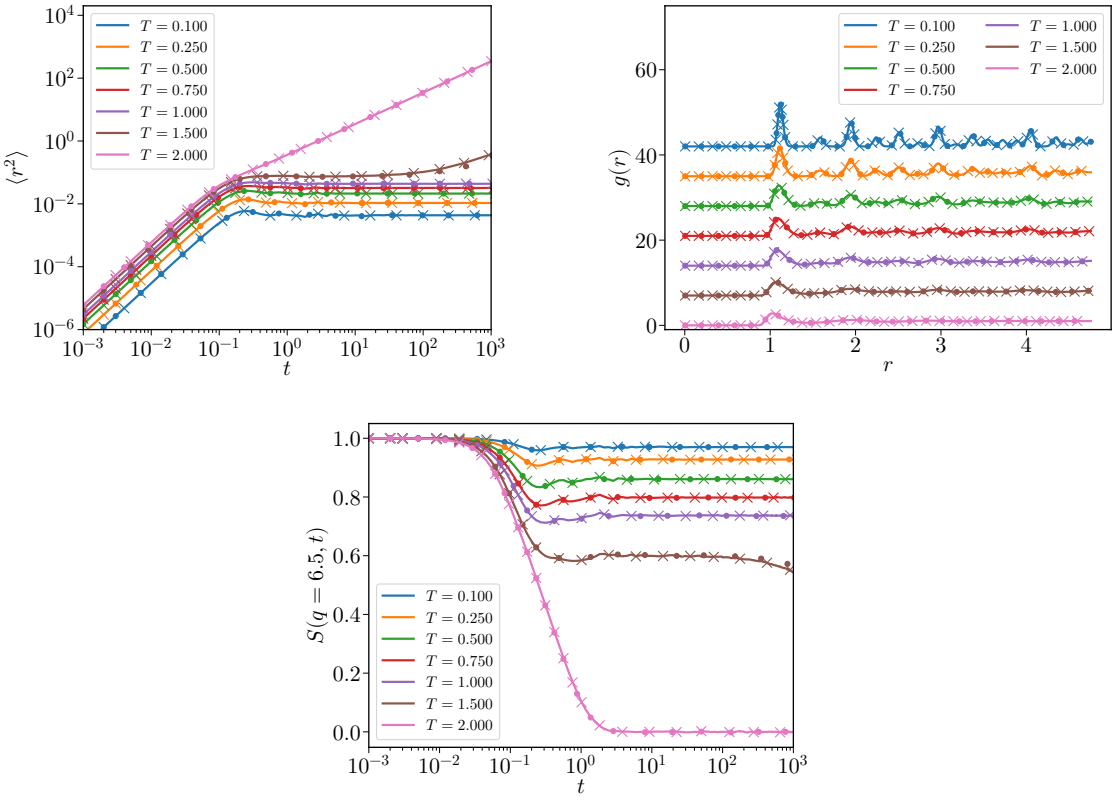


Figure 3.9: Comparison of MSDs (top left), RDFs (top right), and SISFs (bottom) between the force-shifted Lennard-Jones potential (solid lines) and the corresponding neural network potentials trained on the full dataset (crosses) and on dense configurations only (dots), with force weight $\mathcal{W}_\mathrm{F} = 10.0$, at density $\rho = 1.0$ and different temperatures T. RDFs are shifted for clarity.

Summary

The results presented here demonstrate that the LJ potential can be reproduced well even by the relatively simple HDNNP architecture. Without the need for elaborate dataset design or extensive hyperparameter tuning, the networks are able to accurately capture both energies and forces across a wide range of the phase-space. This shows the robustness of NNPs when applied to systems governed by simple pairwise interactions.

An interesting observation was made for the energy-shifted LJ potential. While the reference model exhibits a jump in the forces at the cut-off, the trained NNPs smooth out this discontinuity. This follows naturally from the design of the descriptors, which enforce continuity by construction. The trade-off is a slight loss of accuracy in reproducing forces near the cut-off. Although such jumps are unphysical and their removal can be seen as an improvement, this also highlights that the energy decomposition made by NNPs may alter fine details compared to the reference system.

Having established that NNPs can reliably reproduce the structural and dynamical properties of LJ fluids and crystals, the next step is to increase the complexity of the model system. In the following section, the binary Kob–Andersen mixture is investigated, a model specifically designed to suppress crystallization and capture glassy dynamics.

3.2.3 Binary Kob–Andersen Mixture

To study systems with slow dynamics and glass-forming capabilities, binary mixtures introducing size-polydispersity are investigated. The binary Kob–Andersen (KA) Lennard-Jones mixture of large A and small B particles is an extensively studied model system [21, 77]. The Lennard-Jones interaction parameters are specifically chosen to avoid crystallization. The system is typically defined at a concentration of $\chi_\mathrm{A} = 0.8$ large particles and number density $\rho = 1.2$.

The Lennard-Jones parameters for the KA mixture are given by [21]

$$\epsilon_\mathrm{AA} = 1.0\,\epsilon, \qquad \sigma_\mathrm{AA} = 1.0\,\sigma, \tag{3.18}$$

$$\epsilon_\mathrm{BB} = 0.5\,\epsilon, \qquad \sigma_\mathrm{BB} = 0.88\,\sigma, \tag{3.19}$$

$$\epsilon_\mathrm{AB} = 1.5\,\epsilon, \qquad \sigma_\mathrm{AB} = 0.8\,\sigma. \tag{3.20}$$

The interactions are truncated and force-shifted, with cut-off radii $r_\mathrm{cut,AA} = 2.5\sigma$, $r_\mathrm{cut,BB} = 2.2\sigma$, and $r_\mathrm{cut,AB} = 2.0\sigma$. While this additional force-shift alters the phase boundaries slightly compared to the original formulation, it guarantees smooth forces at the cut-off. Again, reduced units are employed with $\epsilon = 1.0$ and $\sigma = 1.0$.

Upon cooling, the KA mixture displays characteristics of a fragile glass former [81]. Despite its simple interaction, it captures many features of glassy dynamics while remaining computationally inexpensive, making it ideal to investigate the ability of NNPs to reproduce slow relaxation, where even subtle differences in the learned PES could lead to significant deviations in the dynamics.

Network Training

In terms of network architecture and training procedure, the setup is similar to that of the single-component Lennard-Jones system, discussed in sec. 3.2.2. The only substantial difference is the presence of two species, which requires the construction of separate networks for A and B particles. For the A network, symmetry functions are constructed for AA and AB interactions, with the resulting values concatenated into the input vector. Analogously, the B network is provided with descriptors constructed from BB and BA interactions.

The training dataset consists of 25 000 configurations, each containing 500 particles at density $\rho = 1.2$ and concentration $\chi_\mathrm{A} = 0.8$. Temperatures are sampled uniformly across the range $T \in [0.3, 1.2]$. Restricting the dataset to a single density and concentration reflects the standard definition of the KA mixture. Despite the limited dataset, it provides sufficient coverage of the relevant configuration space.

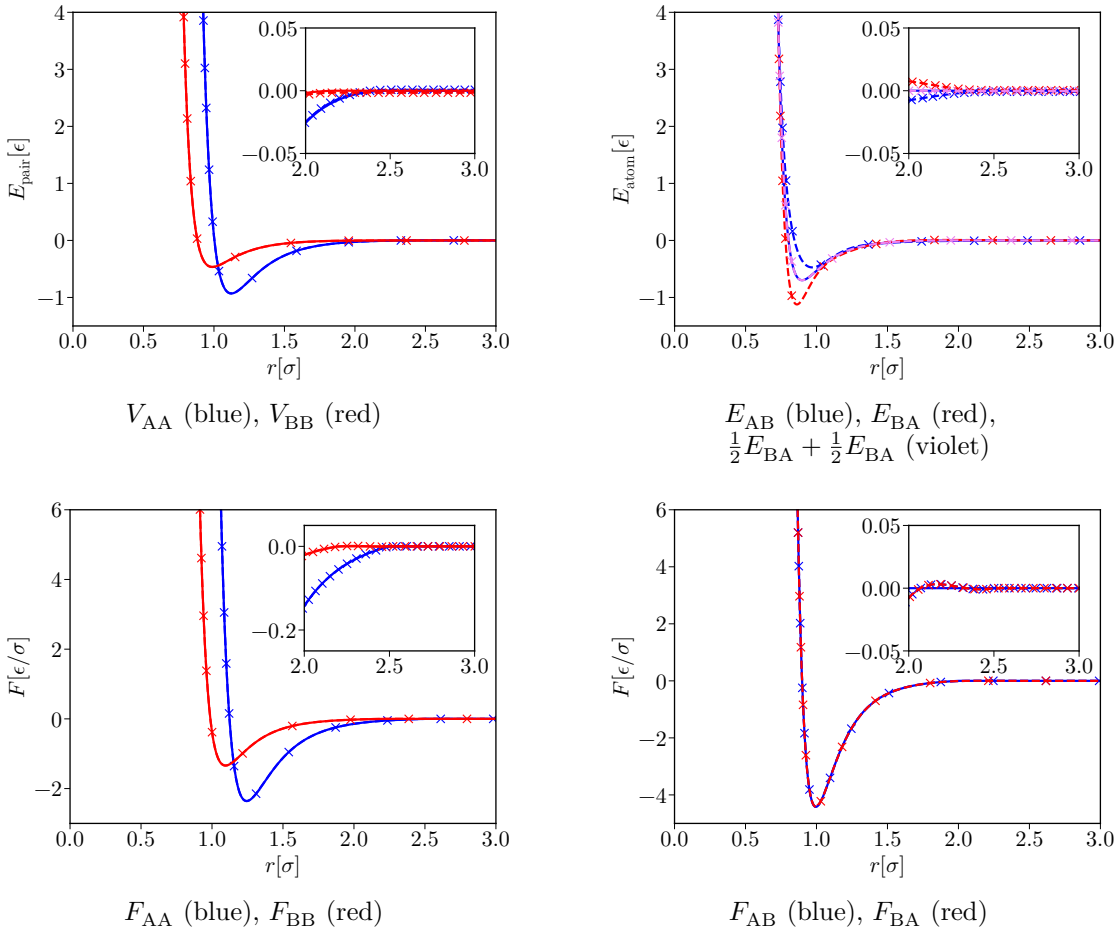
Although the true cut-off distances of the pair interactions differ across species, all descriptors are constructed with a common cut-off radius of $r_\mathrm{cut} = 2.5$. This uniform choice ensures consistency in descriptor construction. Training with n2p2 proceeds without difficulty, yielding energy and force errors comparable to those achieved for the single-component LJ system in sec. 3.2.2. For this reason, training results are omitted here.

One possible problem arises from the comparatively small number of B particles (20% of the total), which may lead to insufficient coverage of the local environments of B particles. This deficiency is partially compensated by the larger system size, 500 atoms compared to 256 in the single-component case. The best-performing NNP is obtained using the same training parameters as used for the LJ-system, i.e., force weight $\mathcal{W}_\mathrm{F} = 10.0$ in the loss.

Potential Evaluation

The quality of the trained NNP for the KA mixture can be evaluated by comparing the effective interactions for AA-, AB-, and BB-pairs. An interesting subtlety arises from the architecture itself. Since two distinct networks are used to predict the atomic energies of A and B particles, the symmetry between the AB and BA interactions is not enforced. Instead, the networks are trained based only on total energies, allowing deviations in the individual AB and BA contributions as long as their sum yields the correct energy.

The AA and BB pair-potentials are reproduced with high accuracy, see fig. 3.10, even in the case of the BB interaction, whichs smaller cut-off radius is learned approximately by the network. For the AB and BA contributions, the atomic energies predicted by the A and B networks deviate from their reference pair potential, but the sum matches the energy of the base model. The pair-potential even vanishes smoothly at the correct cut-off distance despite the network having a larger cut-off radius defined than the underlying interaction. The corresponding forces are reproduced well, since they result from the sum of the gradients. Minor deviations appear near the cut-off, probably a consequence of training only on dense configurations in which isolated AB or BA pairs are not contained.



V_{AA} (blue), V_{BB} (red)

E_{AB} (blue), E_{BA} (red), $\frac{1}{2}E_{\mathrm{BA}} + \frac{1}{2}E_{\mathrm{BA}}$ (violet)

F_{AA} (blue), F_{BB} (red)

F_{AB} (blue), F_{BA} (red)

Figure 3.10: Comparison of pair energies and corresponding forces between the Kob–Andersen mixture and its trained neural network potential for the different species interactions as a function of distance r. Solid lines represent the Lennard-Jones reference potentials, while dashed lines with crosses indicate the neural network predictions. For the symmetric AA and BB pairs the pair potential is shown directly, whereas for the mixed AB and BA pairs the atomic energy contributions of the respective networks are plotted.

Interestingly, and in contrast to the single-component LJ system trained on dense data only, the KA mixture does not show a (significant) shift of energies. This difference could be attributed to the binary composition of the system, where even at high density, a given particle can have relatively few neighbors of the same species, particularly in the case of the minority B particles. This naturally introduces some "dilute" local environments into the training data.

As in the single-component case, the quality of the potential is also assessed through microcanonical energy conservation by comparing the discretization error of the total energy for different discretization steps. Results are shown in fig. 3.11. Since the force-shifted LJ potential is used as reference model, the KA mixture exhibits the expected Δt^2 scaling. The NNP reproduces this behavior exceptionally well, despite employing a descriptor cut-off larger than the cut-offs of the reference interactions.

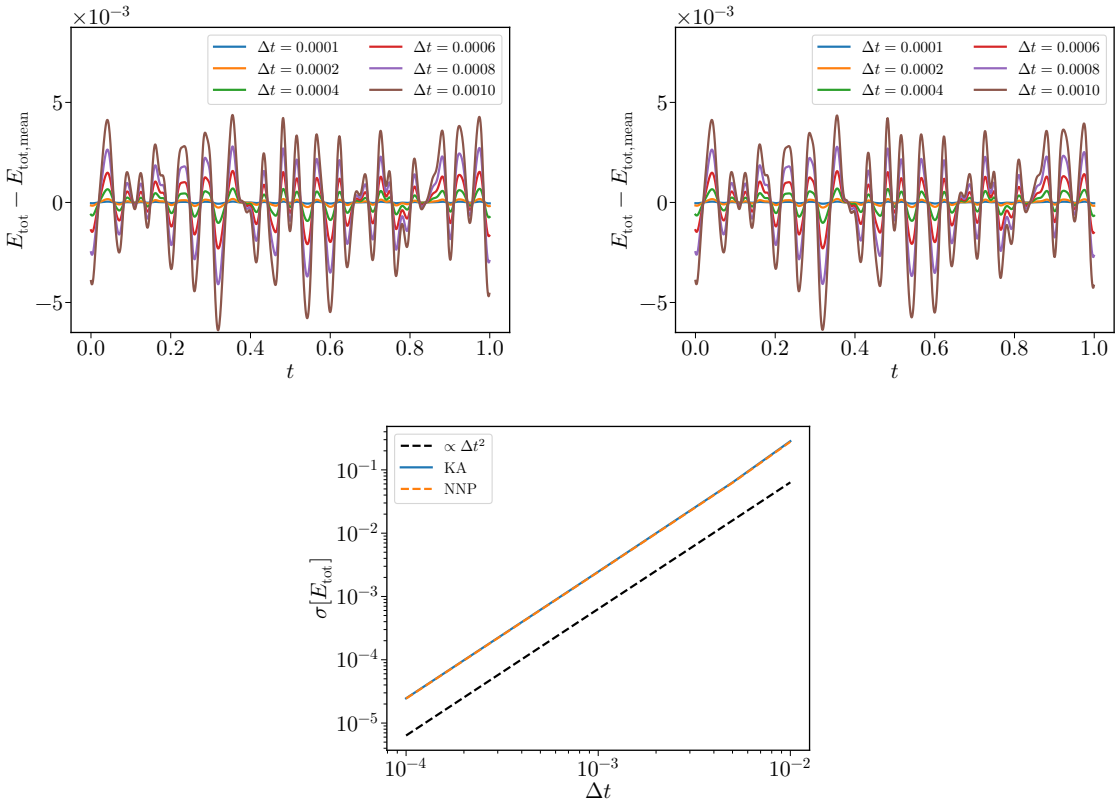


Figure 3.11: Microcanonical energy fluctuations of the Kob–Andersen mixture (top left) and the corresponding neural network potential (top right) at $T = 1.0$ for different timesteps Δt. The bottom panel shows the standard deviation of the total energy E_{tot} as a function of Δt, demonstrating excellent agreement with the expected scaling behavior.

It should be noted, however, that poorly trained potentials may show different behavior. If excessive extrapolation is required for a configuration, conservation of energy is no longer

guaranteed, and systematic drift may appear. For the relatively simple LJ interaction, such failures are difficult to induce even when training on small or incomplete datasets, since the networks usually generalize well in this setting. Here, even the energy fluctuations follow the same pattern as the reference model across the investigated temperature range. While this agreement is consistent at both high and low temperatures, it becomes less visually striking in the regime of slow dynamics near the glass transition, where energy fluctuations themselves are small.

Simulation Comparison

To assess whether the trained NNP can also reproduce dynamical behavior of the Kob–Andersen mixture, simulations of the reference model are compared to simulations with the NNP. In order to improve statistics, particularly for the minority B particles, the total number of particles is increased compared to the single-component LJ system. Four independent simulations are conducted for each state point with 2000 particles each. The simulations are started from randomized high-temperature configurations and subsequently equilibrated in the canonical (NVT) ensemble at target temperature T. Special care was taken for equilibration at low temperatures.

After equilibration, production runs are carried out in the microcanonical (NVE) ensemble. To further enhance statistics, sub-ensembles are sampled from each run. For the low-temperature system, where relaxation times are very long, sufficient temporal separation between sub-ensembles is ensured to avoid significant correlations between starting points.

The static structure is evaluated via radial distribution functions, shown in fig. 3.12, although the corresponding static structure factors (not shown) confirm the same level of agreement. Dynamical properties were probed by the self-intermediate scattering functions, displayed in fig. 3.13, and by the mean squared displacement (not shown), which yields consistent results. Across the entire temperature range, only minor deviations are observed between the NNP and the reference system. These deviations are within the expected level of statistical uncertainty given the finite number of ensembles and the inherent difficulties of equilibrating glassy systems.

Particular care must be taken when interpreting observables involving BB correlations, since B particles represent only a fraction $\chi_{\mathrm{B}} = 0.2$ of the system. Nonetheless, the network reproduces the relevant structural and dynamical correlations, albeit with slightly larger noise compared to AA or AB functions.

Overall, the dynamics of the KA mixture are reproduced extraordinarily well by the NNP, even in the regime close to the glass transition. This is remarkable, given that the network was trained on a single density and composition. Sampling the important temperature

range appears to be sufficient to capture the essential physics, likely reflecting the simplicity of the Lennard-Jones interactions and the limited impact of B particles on the relaxation behavior.

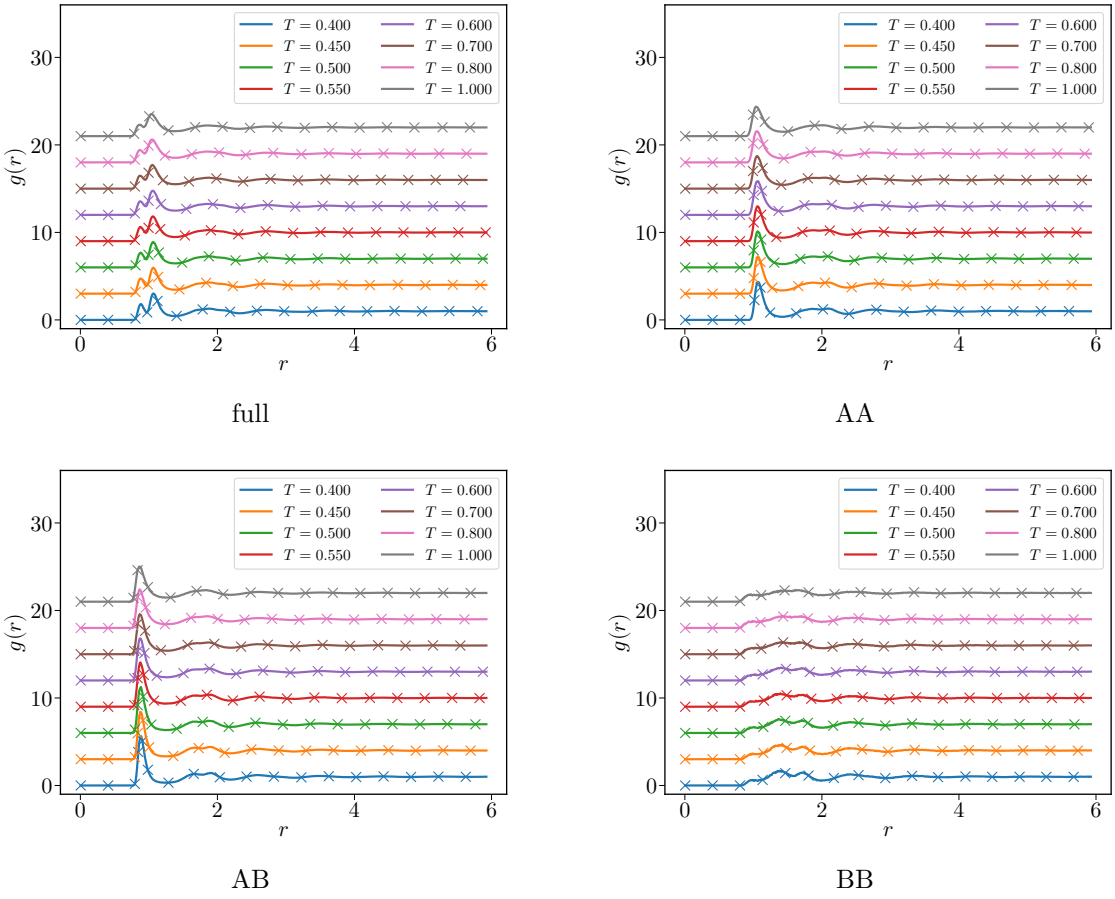


Figure 3.12: Comparison of radial distribution functions (RDFs) of the Kob–Andersen mixture (solid lines) with simulations using the neural network potential (crosses) at different temperatures T. The panels show the total as well as all partial interspecies RDFs.

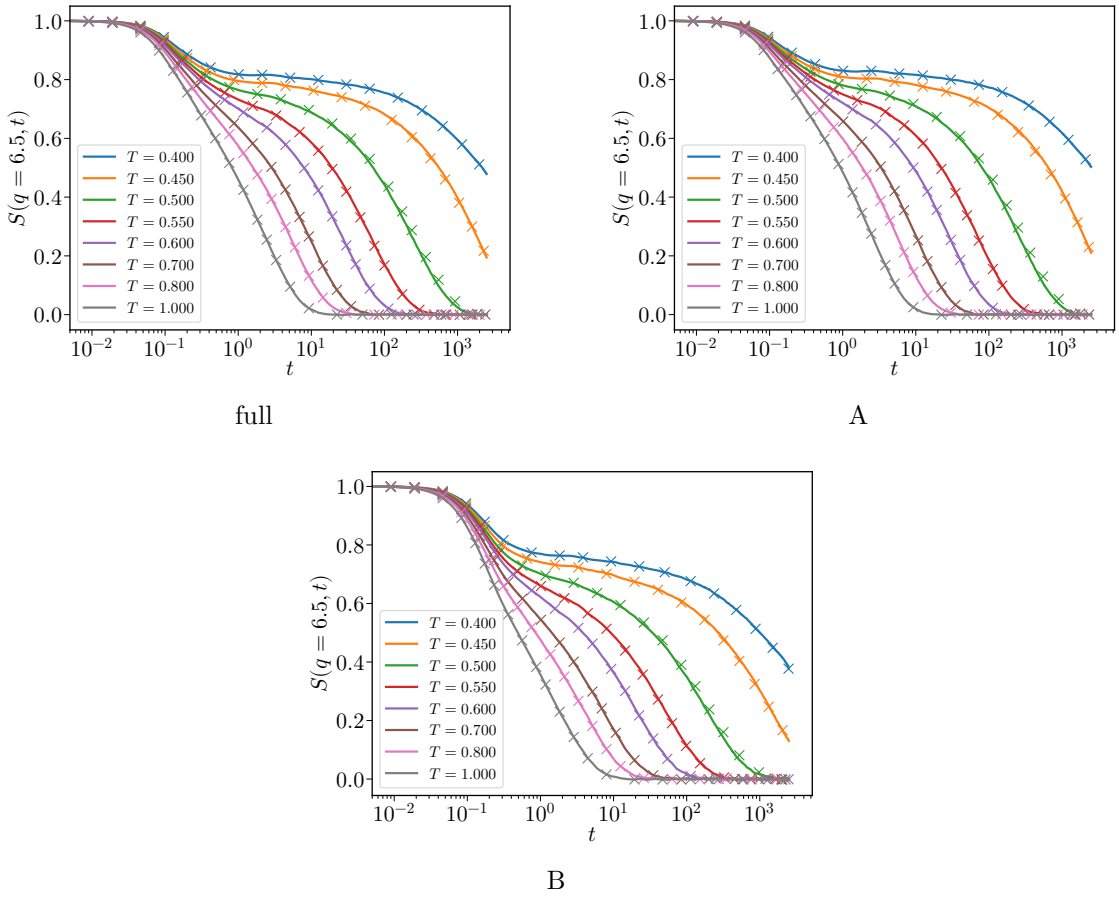


Figure 3.13: Comparison of self-intermediate scattering functions (SISFs) of the Kob–Andersen mixture (solid lines) with simulations using the neural network potential (crosses) at different temperatures T, shown for the total system as well as for the individual species.

Breaking Neural Network Potentials

For the simple LJ interaction, even intuitive parameter choices and straightforward training procedures typically yield neural network potentials of high quality. Guidelines for training NNPs are well established, and for many systems this suffices to obtain models that reproduce both statics and dynamics reliably. Nevertheless, it should be explored how NNPs may fail. One strategy to break their properties is to deliberately restrict the training set to a subset of configuration space and apply the potential to regions of phase-space far from those represented in the training data. This forces the network to the regime of excessive extrapolation, where neural networks are known to perform poorly.

Restricting training data to a small portion of configuration space can occur unintentionally, particularly for complex systems where generating representative training data is not trivial. To emulate this, a NNP is trained solely on low-temperature configurations with $T < 0.6$. Such a dataset captures only limited local environments, as the low kinetic energy at these temperatures prevents sufficient sampling of the strongly repulsive regime. The resulting NNP performs adequately within the phase-space region its trained on but begins to fail for increasing temperatures. Here, extrapolation errors accumulate, which destabilizes the simulations. The consequences are illustrated in fig. 3.15, where the energy fluctuations grow dramatically with increasing temperature, showing unstable dynamics at the highest simulated values. The corresponding fluctuation scaling is depicted in fig. 3.14.

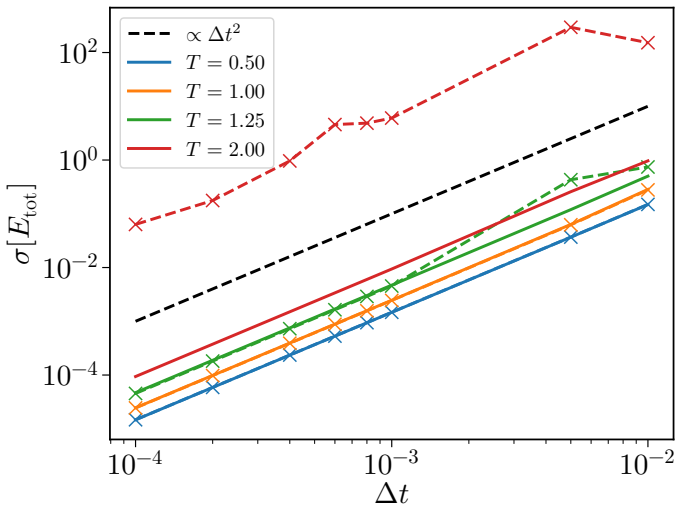
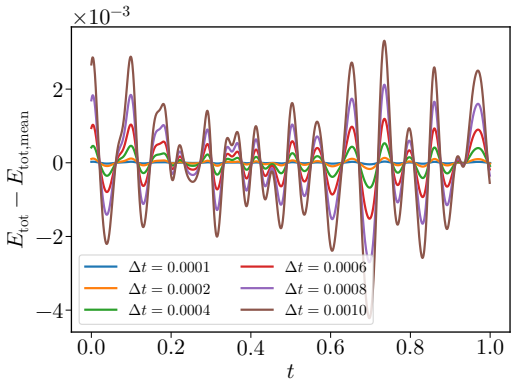
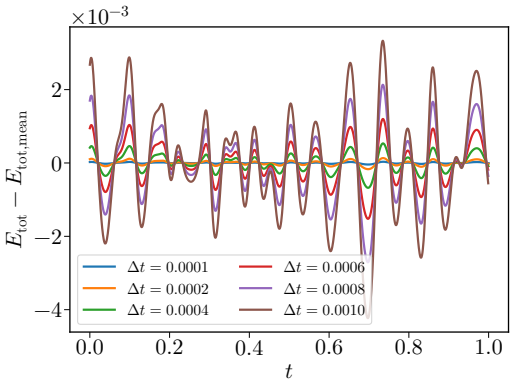


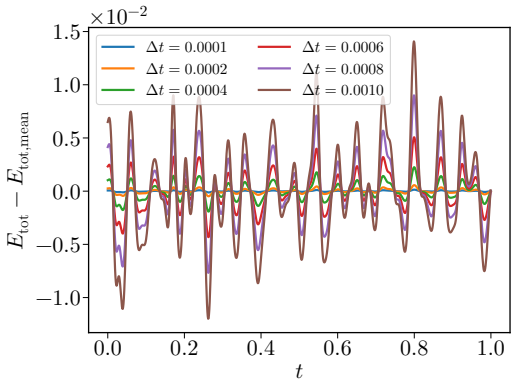
Figure 3.14: Standard deviation of the microcanonical energy fluctuation for the Kob–Andersen mixture (solid) compared to neural network potential simulations (dashed with crosses) at different temperatures T as a function to discretization time-step Δt. The neural network potential was trained exclusively on configurations with $T < 0.6$.



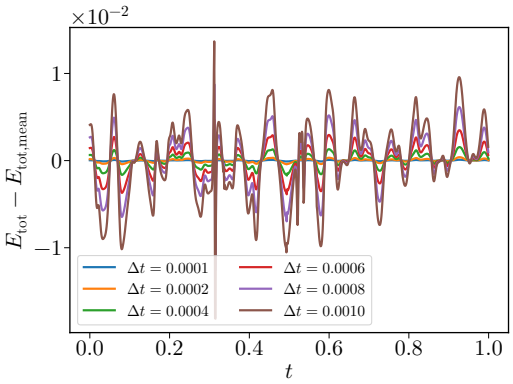
KA mixture ΔE_{tot} for $T = 0.5$



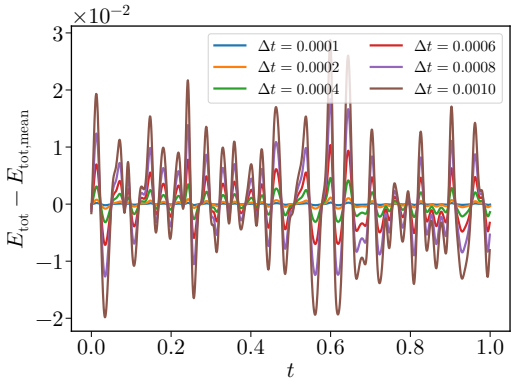
NNP simulation ΔE_{tot} for $T = 0.5$



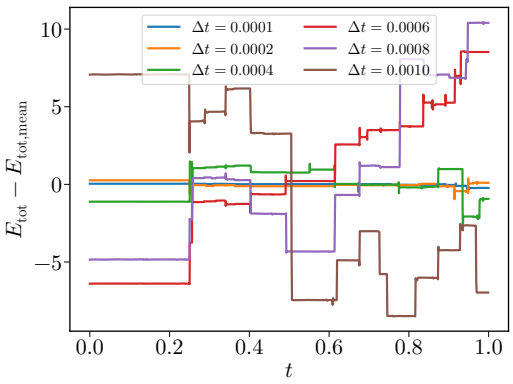
KA mixture ΔE_{tot} for $T = 1.25$



NNP simulation ΔE_{tot} for $T = 1.25$



KA mixture ΔE_{tot} for $T = 2.0$



NNP simulation ΔE_{tot} for $T = 2.0$

Figure 3.15: Microcanonical energy fluctuations of the Kob–Andersen mixture (left) and the corresponding neural network potential (right) at different temperatures T for discretization time-steps Δt. The neural network potential was trained exclusively on configurations with $T < 0.6$.

That higher-temperature simulations can become unstable is not surprising, as seen from the energy fluctuation scaling. The impact is also clearly visible in structural and dynamical observables. In fig. 3.16, the radial distribution functions (RDFs) show large deviations for small interparticle distances, reflecting the failure of the network to learn the short-range repulsion. Similarly, the self-intermediate scattering functions (SISFs) in fig. 3.17 are no longer reproduced once the system is simulated outside the trained temperature range. As the temperature increases, the instability worsens, leading to collapse of the system. Simulations at temperatures below the training threshold remain well reproduced.

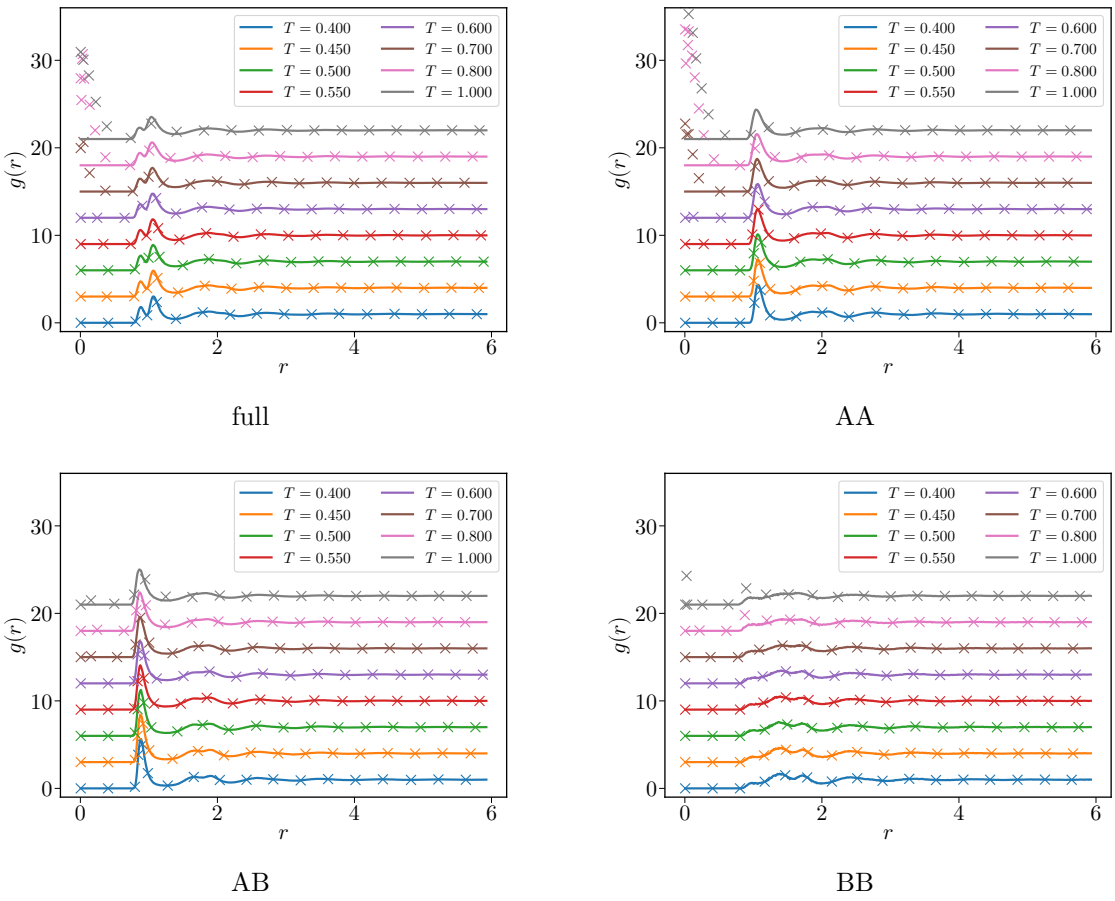


Figure 3.16: Comparison of radial distribution function (RDF) between the Kob–Andersen mixture (solid) and neural network potential simulations trained on only low temperatures $T < 0.6$ (crosses) for different temperatures T. The RDFs for the full system and all species pairs are shown.

While low-temperature datasets fail to sample the configurations space of high-temperature systems, the reverse is not necessarily true. High-temperature datasets naturally contain configurations representative of lower temperatures, albeit at lower probability. Thus, training on high-temperature data generally provides sufficient coverage to reproduce

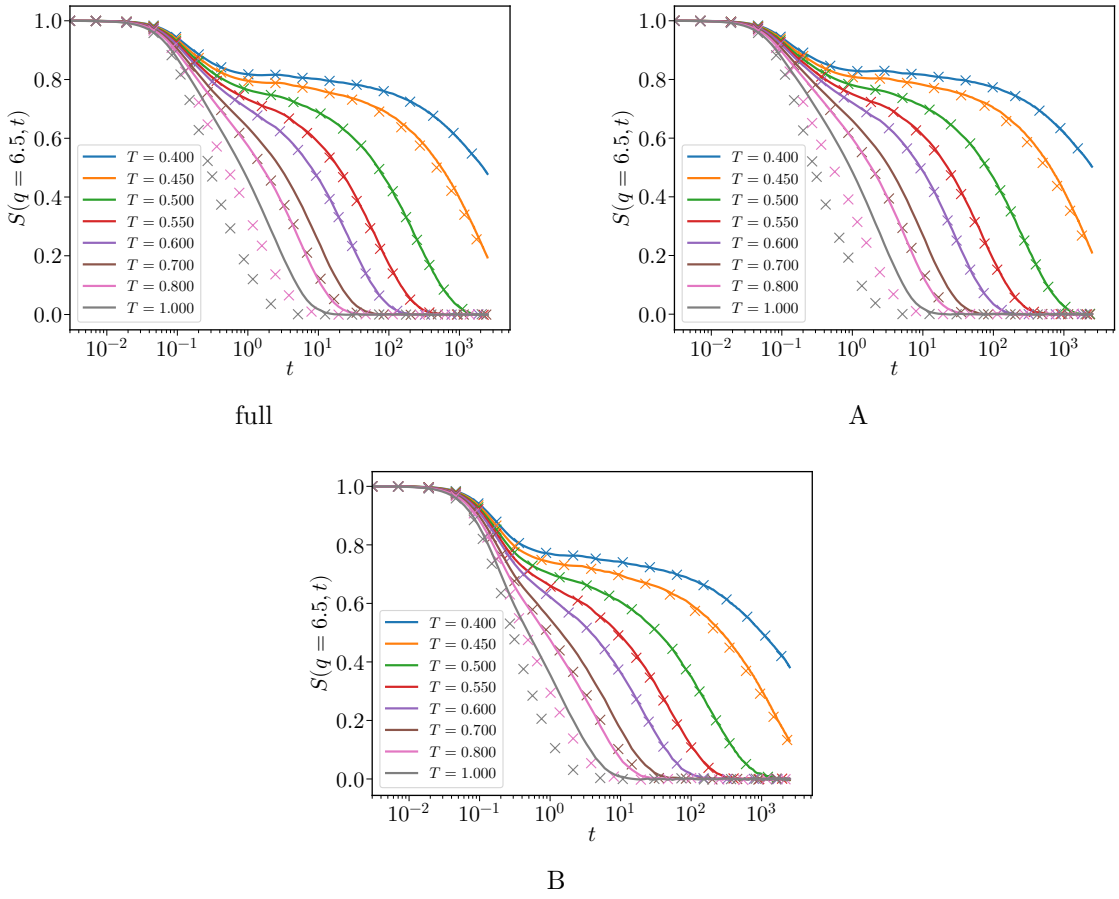


Figure 3.17: Comparison of the self-intermediate scattering functions (SISFs) between the Kob–Andersen mixture (solid) and neural network potential simulations trained on only low temperatures $T < 0.6$ (crosses) for different temperatures T. The different species SISF are shown aswell as for the full system

low-temperature physics for the LJ potential, where the underlying PES is simple and does not qualitatively change across phase-space regions. In more complex systems, however, the PES itself may vary between the different thermodynamic states, and extrapolation across such regions is bound to fail.

These results emphasize that the stability and accuracy of NNPs relies critically on avoiding extrapolation. Ensuring sufficient coverage of relevant configurations is therefore essential, both to guarantee reliable performance and to avoid catastrophic failures when the model is applied beyond its design domain.

Summary

The results presented in this section demonstrate that both the single-component Lennard-Jones system and the Kob–Andersen binary mixture can be reproduced by comparatively simple NNPs, such as the HDNNP architecture. Training requires only a subset of densities and compositions, provided relevant environments are sampled. Established guidelines and standard hyperparameters already ensure stable and accurate training in such simple systems, leaving little room for failure unless the dataset or architecture is deliberately manipulated.

Crucially, reliable reproduction of the underlying potential demands that the training data probes the physically important regions of configuration space. For the LJ system, this requires sampling both repulsive and attractive regimes across relevant length scales. Attempts to force networks into excessive extrapolation, for example by training exclusively on low-temperature configurations and then applying the model at high temperatures, inevitably breaks conservation laws and produces unstable dynamics. For mixtures, additional pitfalls may arise from not covering enough different concentrations, especially when the underlying interactions become more complex.

Overall, it is remarkably difficult to break such simple pairwise potentials without deliberate manipulation. To uncover further challenges and limitations in the context of slow dynamics, more complex interactions should be considered. One such system is the Voronoi model, which features many-body interactions defined by geometric tessellations. This potential cannot be decomposed into pair contributions, making it much closer to the PES of complex materials. At the same time, Voronoi particle simulations remain computationally efficient and scalable. In particular, the binary Voronoi system exhibits glassy dynamics similar to the Kob–Andersen mixture, positioning it as an ideal candidate for testing NNPs on systems with complex many-body interactions while still allowing for direct comparison to the underlying potential.

3.3 Single-Component Voronoi System

Voronoi particles are a model system inspired by the behavior of biological cells and tissue-like materials, where effective interactions emerge from geometric constraints [22, 23]. Each particle is associated with a Voronoi cell, defined as the region of space closer to that particle than to any other. The tessellation into Voronoi cells serves as the foundation for defining an interaction potential. In three dimensions, the potential energy and resulting forces are defined as

$$E_{\mathrm{pot}} = \sum_{i=1}^{N} E_{\mathrm{pot},i} = \frac{\gamma}{2} \sum_{i=1}^{N} \int_{v_i} \mathrm{d}^3\boldsymbol{r}\, r^2, \tag{3.21}$$

$$\boldsymbol{F}_i = \gamma \boldsymbol{\tau}_i = \gamma \int_{v_i} \mathrm{d}^3\boldsymbol{r}\, \boldsymbol{r} = \gamma v_i \left(\boldsymbol{r}_i - \boldsymbol{r}_{\mathrm{c},i}\right), \tag{3.22}$$

where v_i denotes the volume of the Voronoi cell of particle i, $\boldsymbol{r}_i$ its position, and $\boldsymbol{r}_{\mathrm{c},i}$ the centroid of the cell. The parameter γ sets the energy scale, while $\boldsymbol{\tau}_i$ represents the centroidal moment. To set the energy and length scales to values comparable to LJ-systems, $\gamma = 1000$ is chosen. The Voronoi tessellation is efficiently carried out using the Voro++ library [82] and can be integrated into existing simulation frameworks such as LAMMPS[‡]. An example of a Voronoi tessellation in two dimensions is shown in fig. 3.18.

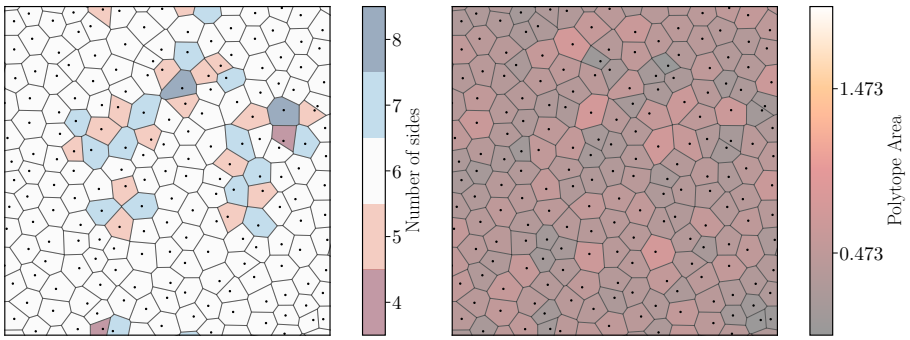


Figure 3.18: Voronoi tessellation of a two-dimensional particle configuration obtained by applying small random displacements to an ideal hexagonal crystal. The tessellation was computed using the freud python-package [83].

The Voronoi interaction is inherently many-body, as the energy of each particle depends non-trivially on the geometry of all neighboring Voronoi cells. There is no distinct pair part of the interaction, and the number of contributing neighbors is determined by the tessellation rather than by a distance cut-off. As a consequence, the PES cannot be decomposed into pairwise additive contributions. This places the Voronoi potential conceptually closer to

[‡]I thank Jean Farago and Olivier Benzerara from Université de Strasbourg for providing the code and for their assistance in setting it up.

interactions encountered in DFT.

The Voronoi system exhibits interesting dynamical behavior, particularly in the supercooled regime. While the single-component fluid already shows complex relaxation processes, binary Voronoi mixtures are excellent glass formers [23]. Increasing the density does not slow down the dynamics as in typical interaction models. Instead, it accelerates relaxation, a counterintuitive effect originating from the geometric nature of the interaction.

From a computational standpoint, the Voronoi potential presents an interesting system for training neural network potentials (NNPs). On the one hand, the PES can be evaluated with high accuracy for arbitrarily large systems at negligible cost, allowing good sampling of the thermodynamically relevant regime, including solid, meta-stable liquid, and liquid states. Exhaustive datasets can be generated without prohibitively expensive reference calculations. On the other hand, the variable number of Voronoi neighbors contributing to the atomic environments complicates the construction of descriptors based on local cut-offs. Any NNP representation has to effectively learn a redistribution of atomic energies consistent with the interaction. Simple feed-forward architectures with conventional distance-based descriptors may struggle to capture the peculiarities of the Voronoi potential. Therefore, this system serves as an ideal test case to evaluate the descriptive power of different neural network architectures, ranging from HDNNPs to more advanced graph-based approaches.

3.3.1 Training Data

Constructing a reliable dataset for NNPs of the Voronoi potential requires careful consideration of both the geometry of the tessellation and the thermodynamic conditions under which configurations are sampled. The many-body nature of the Voronoi potential increases demands placed on dataset coverage. Suitable thermodynamic ranges and cut-off definitions should be investigated to ensure all relevant configurations of the PES are adequately sampled.

Voronoi Neighbor Statistics

A useful starting point is the analysis of Voronoi neighbors statistics. The mean, minimum, and maximum number of neighbors as a function of temperature for different densities are shown in fig. 3.19. While the minimum and maximum values are strongly influenced by rare local configurations, including these outliers is important because they represent possible geometries the model may encounter. As the temperature increases, the distribution of Voronoi neighbor counts broadens. While the minimum decreases, the mean and maximum increase. Raising the density increases the mean number of neighbors only slightly, consistent with the geometric scaling of the tessellation.

For reference, Voronoi tessellations of completely purely random configurations yield an average of $\langle N_{\mathrm{VN}} \rangle \approx 15.54$ neighbors in three dimensions, obtained from the number of vertices reported in [84] combined with Euler's polyhedral formula [85]. The Voronoi systems studied here appear to saturate at a somewhat lower average, likely due to finite-size effects and thermal fluctuations. The saturation of neighbor count suggests that temperatures of $T \lesssim 50$ are sufficient for sampling training data. Including high-temperature states is valuable because they contain rare local configurations that are unlikely to occur but are nonetheless important to ensure stable dynamics.

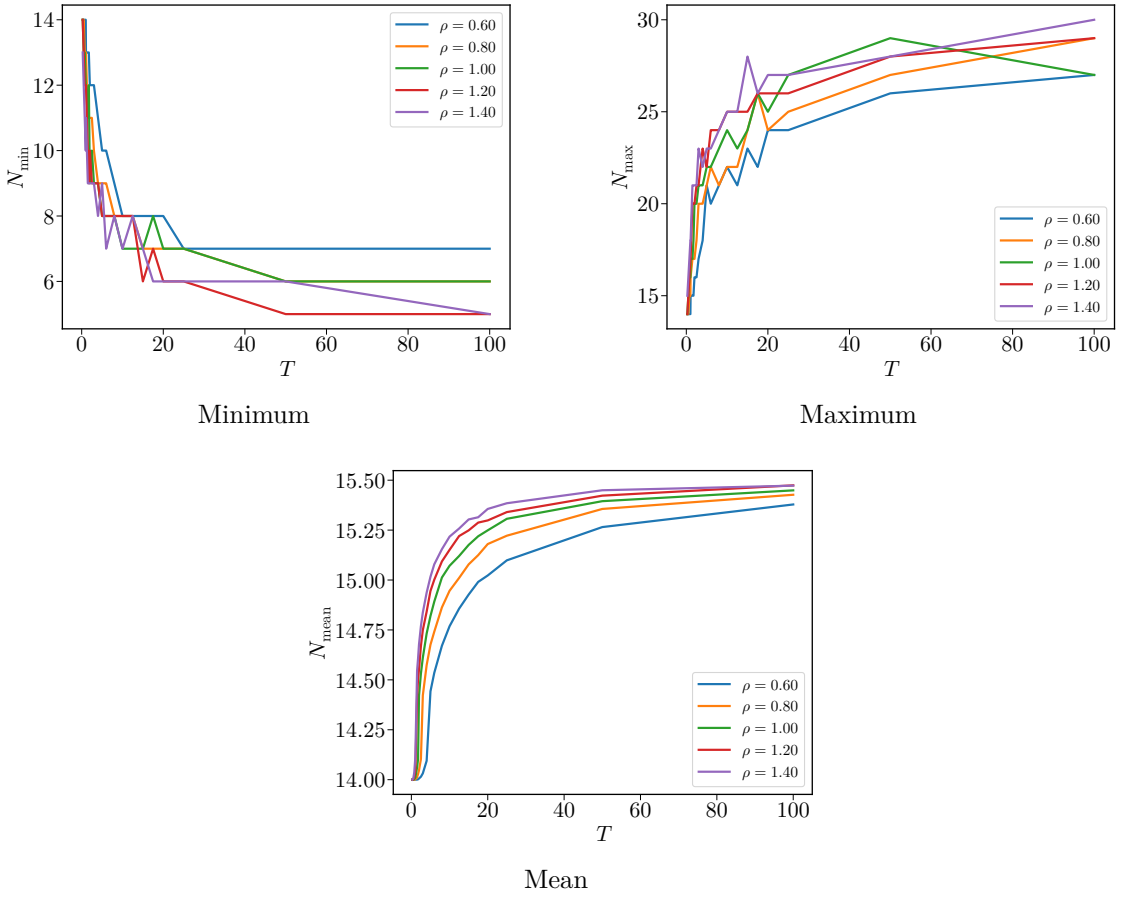


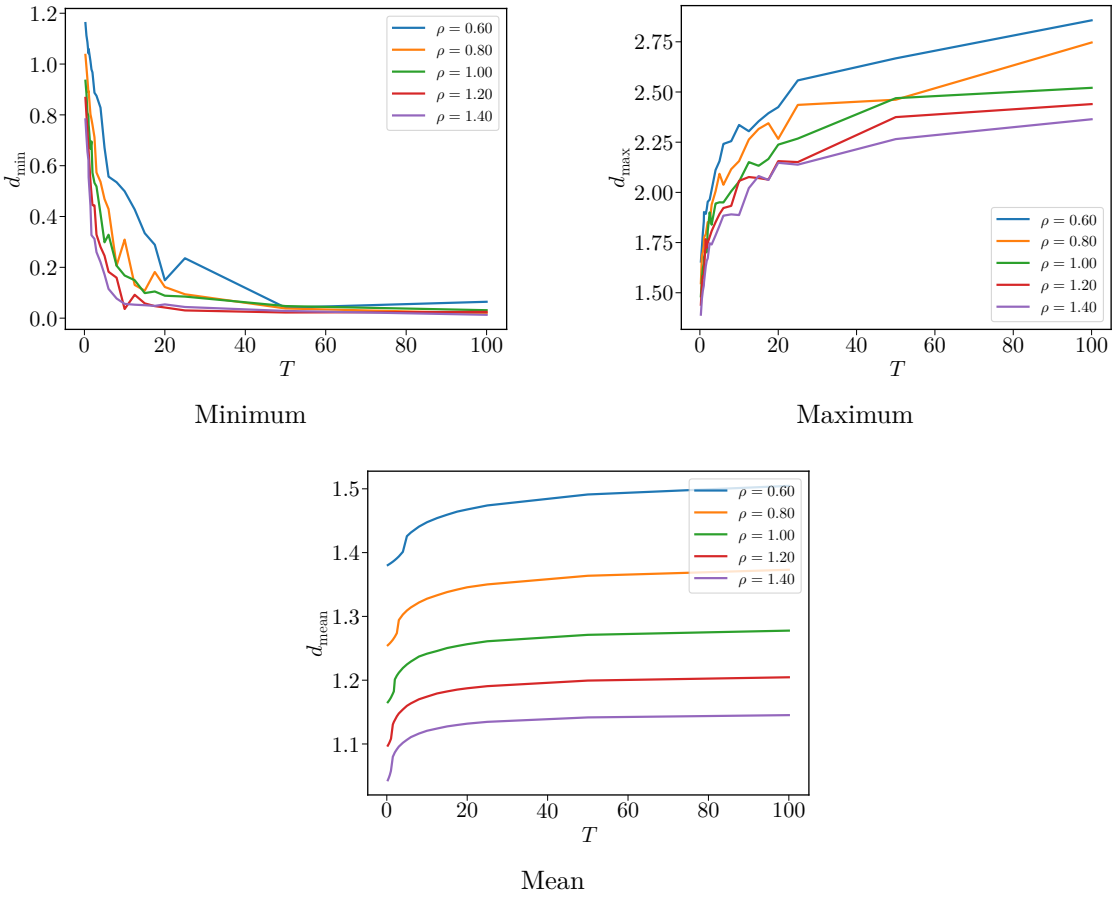
Figure 3.19: Minimum, maximum, and mean number of Voronoi cell neighbors as a function of temperature T for different densities ρ. For each (T, ρ) pair, statistics were obtained by averaging over 200 configurations with 250 particles each.

Neighbor Distances and Cut-Off Estimates

To construct suitable descriptors, an estimate of a suitable cut-off radius is essential. Fig. 3.20 shows the mean, minimum, and maximum distances of a particle to its Voronoi neighbors as a function of temperature. With increasing temperature, the spread of

minimum and maximum neighbor distances increases, while density effects are largely restricted to rescaling of typical lengths. Although a particle only has 14 – 30 Voronoi neighbors, a distance-based cut-off must necessarily include substantially more particles.

From this data, a cut-off $r_{\mathrm{cut}} \approx 2.5 - 3.5$ appears to be required. Tab. 3.3 shows the number of neighbors following from different cut-off radii. This number is close to the number of secondary Voronoi neighbors, although the particles are not necessarily identical. For $T = 0.25$, a mean of about 65 secondary Voronoi neighbors is observed, increasing to nearly 120 at $T = 50.0$. This indicates that a number-based cut-off of $k_{\mathrm{NN}} \approx 96 - 200$ particles would also provide adequate coverage. In either way, it has to be accepted that NNPs will redistribute atomic energies among technically non-contributing atoms, as an exact partition cannot be constructed without performing an (at least implicit) Voronoi tesselation. The resulting dependence on distant particles could be mitigated by large and diverse training sets that statistically suppress these contributions.



Minimum Maximum

Mean

Figure 3.20: Minimum, maximum, and mean distance d between a Voronoi particle and its cell neighbors as a function of temperature T for different densities ρ. For each (T, ρ) pair, statistics were obtained by averaging over 500 configurations with 250 particles each.

r_{cut}	N_{min}	N_{max}	N_{mean}
2.5	43	87	63.1
3.0	88	136	113.0
3.5	152	204	175.8

Table 3.3: Number of atoms within a given cut-off radius around a Voronoi particle in configurations sampled at $\rho = 1.0$ and temperatures $T \in [0.25, 50]$

Dataset

Based on the neighbor analysis, configurations are sampled with temperatures $T \in [0.25, 50.0]$ and densities $\rho \in [0.8, 1.25]$, covering crystalline, meta-stable liquid, and liquid states. The main density of interest here is $\rho = 1.0$, with other densities included to improve stability. The lower bound in temperature is chosen due to rapid crystallization, where particles essentially remain trapped in their Voronoi cells. The upper bound ensures that even rare geometrical configurations are included. Care is taken to sample meta-stable liquids. Starting from randomized configurations, the system is first equilibrated at high temperature in NVT and then cooled to the desired state point. After short relaxation at the desired temperature, configurations are sampled in NVE at fixed intervals.

On average, the dataset has a mean total energy per particle of $E_{\text{mean,pa}} = 132.2$ and a mean force magnitude of $|\boldsymbol{F}|_{\text{mean}} = 79.1$, with many individual force components being close to zero. The relatively large value of the total energy primarily reflects the average size of Voronoi cells. Introducing a constant offset to reduce this magnitude is unnecessary, as most NNPs already include rescaling and shifting of energies within their architecture.

The resulting training sets consist of 50 000 configurations with 250 particles each. This system size is sufficient to avoid finite-size effects in the PES, as interactions are short-ranged by construction. The number of configurations is chosen to ensure an abundance of different local environments in the data. While more sophisticated strategies such as active learning [86] could significantly reduce the required dataset size, such optimizations are not the primary aim here. Instead, the emphasis lies on broad coverage of thermodynamically relevant states, enabled by the computational efficiency of the Voronoi potential compared to DFT-based systems.

Other approaches to dataset generation were also tested. Random configurations, possible here due to the absence of hard-core repulsion, and configurations sampled from LJ systems both proved usable. However, configurations obtained directly from Voronoi simulations consistently produced the most stable and transferable training results, as they optimally probe the relevant phase-space.

3.3.2 Network Training

With a sufficiently broad dataset established and reasonable cut-off values identified, the construction of NNPs for the Voronoi system can proceed. The remaining challenge lies in the choice of how to represent local atomic environments. Different network architectures employ distinct strategies to transform atomic coordinates into invariant descriptors, ranging from hand-crafted functional forms to fully trainable embedding networks. Usually, a more sophisticated descriptor construction improves the descriptive power of the NNP at the cost of computational efficiency. For a system as geometrically complex as the Voronoi potential, simple descriptors are unlikely to suffice, motivating exploration of multiple neural network architectures.

In the following, three different approaches are considered. High-dimensional neural network potentials (HDNNPs), which were successfully applied to the Lennard-Jones system in sec. 3.2, set the baseline. Beyond this, the DeePMD framework [15, 67] is employed, which eliminates the explicit choice of hand-crafted symmetry functions by constructing descriptors through a trainable embedding network. A custom DeePMD-inspired implementation in PyTorch [53] is additionally tested, introducing a cut-off based on the number of nearest neighbors, reflecting the fact that the Voronoi potential does not naturally suggest a radial cut-off. Finally, the NequIP architecture [17, 70], based on graph-neural networks, is considered. This approach is conceptually well aligned with the geometric nature of the Voronoi interaction, as it encodes higher-order angular correlations and tensorial features of the local environments.

As in most applications of NNPs, the exact choice of network parameters requires empirical tuning. Fortunately, the general guidelines provided by the developers of the respective codes often yield hyperparameters that perform reliably, requiring only minimal tuning. Further hyperparameter adjustments primarily aim to balance generalization against overfitting. Excessively small networks fail to capture the many-body correlations, whereas excessively large ones risk poor generalization to unseen data.

Although a wide range of strategies exists to improve the efficiency of training, the need for such methods is circumvented here by employing large and diverse training sets. This allows to focus on the relative performance of different network architectures, instead of having to optimize the training pipeline itself.

High-Dimensional Neural Network Potentials

First, the Voronoi potential is learned using the n2p2-package [12, 64], employing the high-dimensional neural network potential (HDNNP) architecture introduced in sec. 3.1.1. While successfully applied to the Lennard-Jones system, see sec. 3.2, the Voronoi system provides a significant challenge due to its geometry. To adequately capture the complex local environments of Voronoi cells, a substantially larger set of symmetry functions is required, including both radial and angular terms.

A set of 72 symmetry functions is used, consisting of 24 radial and 48 angular functions, generated according to [63] within the cut-off radius r_{cut}. Networks trained with much less symmetry functions failed to converge reliably, whereas increasing the number beyond this choice mainly led to longer training times. The final network architecture comprised two hidden layers with 50 and 25 nodes, respectively, using tanh activation functions. While smaller networks show similar accuracy, the larger network size is preferred as it achieves more consistent convergence during training. Training is performed with both energies and forces in the loss function, with a relative force weight of $\mathcal{W}_{\text{F}} = 10$. Optimization parameters are kept at the recommended defaults [64]. For these settings, the loss converges quickly for both energy and forces, though the final errors remain comparatively large.

A comparison of training histories for different cut-off radii is shown in fig. 3.21, with the corresponding error metrics on the test sets summarized in tab. 3.4. The cut-offs $r_{\text{cut}} = 3.0$ and $r_{\text{cut}} = 3.5$ perform similarly within the noise expected from neural-network training, whereas the smallest cut-off, $r_{\text{cut}} = 2.5$, yields the weakest performance. This indicates that a sufficiently large cut-off radius is required to capture all relevant Voronoi neighbors, but extending the cut-off beyond that range does not improve performance and may even introduce redundant or noisy information that degrades training quality and slows convergence. Although larger cut-offs improve the network performance, the relative force errors encountered here (7% – 10%) may prove problematic for the stability and accuracy of simulations.

r_{cut}	$E_{\text{RMSE,pa}}$	F_{RMSE}	$F_{\text{RMSE}}/F_{\text{mean}}$
2.5	0.08108	7.93472	0.10033
3.0	0.05431	5.72958	0.07245
3.5	0.05672	5.87020	0.07423

Table 3.4: Comparison of HDNNPs for the Voronoi potential with different cutoff radii r_{cut}. The table reports the resulting energy and force errors on the testset, as well as the relative force error with respect to the mean force magnitude.

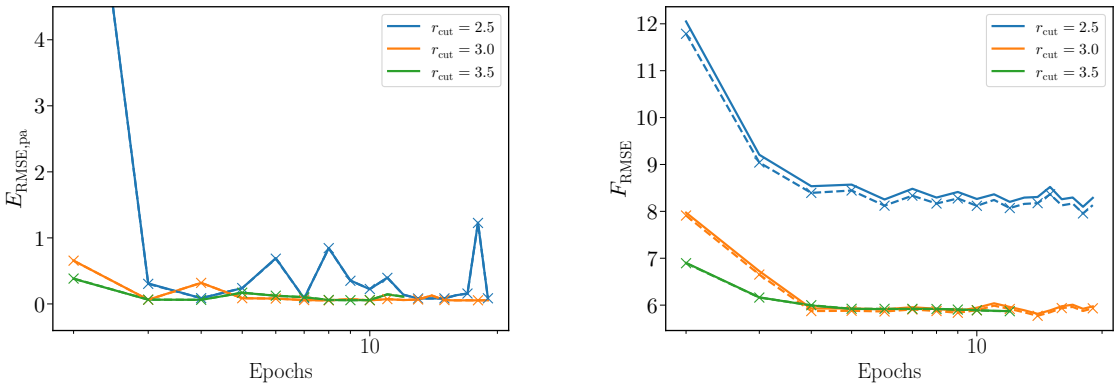


Figure 3.21: Training history of HDNNPs for the Voronoi potential using the n2p2 package with different cutoff radii r_{cut}. The plots show the energy errors (left) and force errors (right) as functions of the training epoch. Solid lines represent training errors, while dashed lines denote validation errors. Training errors are computed as running averages within each epoch.

Deep Potentials

To assess whether the relatively poor training performance of HDNNP comes from limitations of the descriptors, the Voronoi potential is also trained using the Deep Potential framework [15, 16, 66]. The architecture was introduced in sec. 3.1.2. Local environment descriptors are constructed through a fully trainable embedding network. Since the same training dataset is employed, the principal differences should arise from the descriptor construction and the training procedure. DeePMD uses ADAMW optimization of the loss function with an exponentially decaying learning rate.

Given the computational cost of the more elaborate architecture, only limited hyperparameter optimization was performed, guided by the recommendations of the developers [67]. The final networks balance efficiency and accuracy with the following setup:

- An embedding network of three layers with 40 nodes each to construct $G^{i,1}$, with the first 16 columns selected for $G^{i,2}$ as a submatrix of $G^{i,1}$, using tanh activations.
- A fitting network of three layers with 25, 50, and 100 neurons, also with tanh activations, arranged in a residual architecture.
- Training optimization with both energies and forces, and a relative force weight $\mathcal{W}_{\mathrm{F}} = 10$.

Larger networks show only marginal improvements on the validation set but tend to overfit on the training data, while smaller networks peformed worse.

The training histories for different radial cut-off choices ($r_{\text{cut}} = 2.5, 3.0, 3.5$) are shown in fig. 3.22, with corresponding error metrics on the test set summarized in tab. 3.5. Training convergence was noticeably slower and more erratic compared to n2p2, requiring a far larger number of epochs. The best force accuracy was obtained with the largest cut-off, $r_{\text{cut}} = 3.5$, whereas energy errors are comparatively insensitive to the cut-off. The smallest cut-off, $r_{\text{cut}} = 2.5$, performed particularly poorly. Ultimately, the DeePMD-networks fails to outperform the "simpler" HDNNPs, at least for the Voronoi potential, despite their formally more expressive descriptor construction.

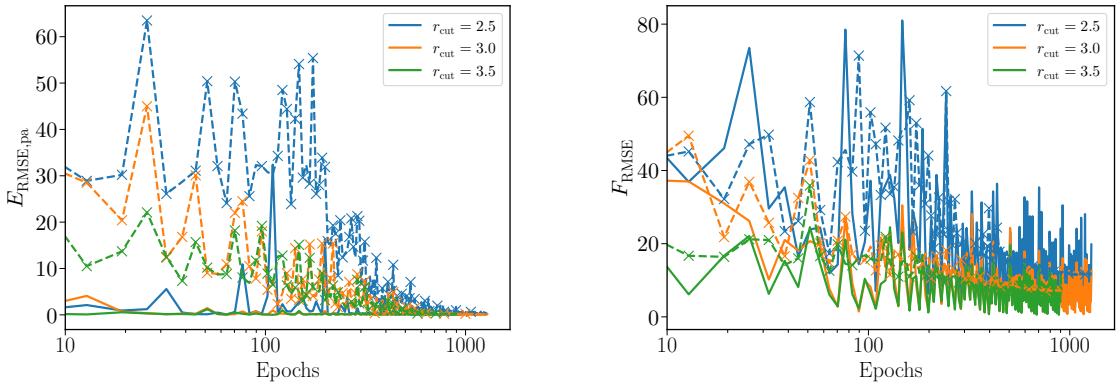


Figure 3.22: Training history of Deep Potentials for the Voronoi potential using the DeePMD package with different cutoff radii r_{cut}. The plots show the energy errors (left) and force errors (right) as functions of the training epoch. Solid lines represent training errors, while dashed lines denote validation errors. Training errors are computed as running averages within each epoch.

r_{cut}	$E_{\text{RMSE,pa}}$	F_{RMSE}	$F_{\text{RMSE}}/F_{\text{mean}}$
2.5	0.22	11.40	0.14
3.0	0.13	6.86	0.09
3.5	0.18	6.02	0.08

Table 3.5: Comparison of Deep Potentials for the Voronoi potential with different cutoff radii r_{cut}. The table reports the resulting energy and force errors on the testset, as well as the relative force error with respect to the mean force magnitude.

An important consideration for geometric interactions such as the Voronoi potential is the definition of the universal radial cut-off. Unlike traditional atomic interactions, the Voronoi interaction does not necessarily decay with distance, but instead depends on topology. Thus, a number-based cut-off, defined by including the k_{NN} nearest-neighbors in the local environment of each particle, is tested as an alternative to the radial criterion.

To explore this idea, the $se_{\text{e2,a}}$ descriptor is modified and reimplemented in a custom PyTorch code [53]. The construction of $\tilde{R}^i$ is limited to the k_{NN} nearest neighbors rather

than all atoms within r_{cut}. The nearest-neighbor search is performed by PyTorch3D [87]. Additionally, the weighting function is simplified to $s(r_{j,i}) = r_{j,i}$, removing the reciprocal form of the original definition. The remaining architecture is kept as close as possible to the official implementation[‡].

The motivation for this modification is straightforward, as a radial cut-off includes significantly more particles than required for the Voronoi tesselation. By contrast, the k_{NN} scheme risks including very distant neighbors in less dense systems. Since only relatively dense states are of interest here, this tradeoff is deemed acceptable. Training employed the same hyperparameters as the radial cut-off network, as alternative settings failed to show noticeable improvement.

Training histories for $k_{\text{NN}} = 96,\ 128,\ 200$ are shown in fig. 3.23, corresponding approximately to the average neighbor counts for $r_{\text{cut}} = 2.5,\ 3.0,\ 3.5$, respectively (compare tab. 3.3). The corresponding test errors are summarized in tab. 3.6. The number-based cut-off achieves performance comparable to the radial cut-off method. Interestingly, the smallest neighbor cut-off, $k_{\text{NN}} = 96$, performed markedly better than the smallest radial cut-off, suggesting improved stability for limited neighborhoods.

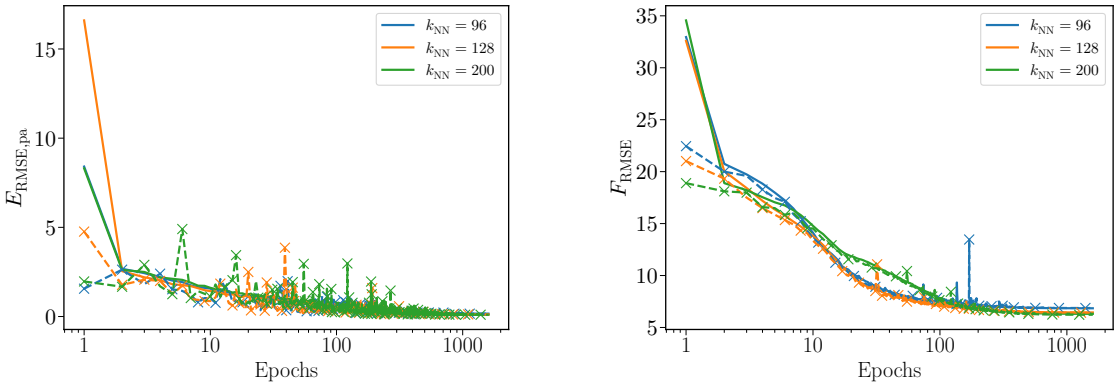


Figure 3.23: Training history of Deep Potentials for the Voronoi potential using the custom k-nearest-neighbor cut-off with different neighbor numbers k_{NN}. The plots show the energy errors (left) and force errors (right) as functions of the training epoch. Solid lines represent training errors, while dashed lines denote validation errors. Training errors are computed as running averages within each epoch.

Implementation differences between PyTorch and TensorFlow, possibly in the details of the ADAMW optimizer, may also contribute to the observed variations. Nevertheless, the number-based scheme is at least computationally more efficient, since the radial method must always accommodate the maximum possible number of neighbors. While training quality remained below that of the HDNNPs, the stability of the k_{NN} approach highlights the potential of investigating alternative cut-off strategies.

[‡]Since completion of this work, a PyTorch backend has been added to the official code.

k_{NN}	$E_{\mathrm{RMSE,pa}}$	F_{RMSE}	$F_{\mathrm{RMSE}}/F_{\mathrm{mean}}$
96	0.15607	6.87274	0.08690
128	0.11459	6.42097	0.08119
200	0.09880	6.22285	0.07868

Table 3.6: Comparison of Deep Potentials for the Voronoi potential using the custom k-nearest-neighbor cut-off with different neighbor numbers k_{NN}. The table reports the resulting energy and force errors on the test set, as well as the relative force error with respect to the mean force magnitude.

Although both radial and number-based cut-off Deep Potentials fail to outperform the HDNNPs for the Voronoi system, they show important methodological aspects. The radial cut-off scheme struggles with convergence for small values of r_{cut}, while the number-based scheme avoids this failure but introduces new tradeoffs for dilute environments. The descriptive power of the network architectures is similar in principle, suggesting that performance differences are primarily tied to training stability and optimization and not the expressiveness of the descriptor. The practical advantage of DeePMD is its efficient GPU implementation compared to the single-threaded, CPU-bound n2p2 framework.

Neural Equivariant Interatomic Potentials

Since both HDNNPs and Deep Potentials showed relatively poor performance, a more descriptive architecture is investigated. The NequIP framework [17, 68] is based on graph-neural networks and has demonstrated superior accuracy for complex systems where higher-order correlations are essential to represent local atomic environments. Such a construction is particularly appealing for the Voronoi potential. The graph-based representation of NequIP naturally aligns with this system, with atoms interpreted as graph nodes and interatomic connections as edges.

The architecture of NequIP was introduced in sec. 3.1.3. Among all tested neural network potentials, it is by far the most computationally expensive, requiring millions of trainable parameters, in contrast to the several hundreds or couple thousands used in n2p2 or DeePMD. Consequently, hyperparameter selection plays an important role. A complete list of all tunable parameters would be excessive[‡], and only the most relevant choices are highlighted here:

[‡]The example input file for network construction already specifies more than 20 tunable parameters for the baseline setup alone.

- Five interaction blocks are used, as they provide the most consistent overall performance.
- The maximum rotational order l_{max}, which determines the highest-order tensorial features retained in the interaction blocks, is systematically varied.
- The feature multiplicity within the interaction blocks is set to 32, controlling the mixing of features with the same rotational order and parity.
- The radial network in the convolutional filters employs a Bessel basis with eight functions and a three-layer residual architecture with 64 neurons per layer.

The ADAMW optimizer is used together with cosine annealing of the learning rate[‡], and training is carried out using a relative force weight $\mathcal{W}_{\text{F}} = 10.0$. In cosine annealing, the learning rate is reduced according to a half-cosine schedule whose period increases exponentially over the course of training.

While the number of layers and neurons primarily controls network size and efficiency, the most influential hyperparameters are l_{max} and the cut-off radius r_{cut}. Moderate tuning of the remaining hyperparameters typically led to only minor improvements, often at the cost of efficiency.

Several networks with different rotational orders and cut-off radii are trained on the dataset. Training histories for $l_{\text{max}} = 1$ and $l_{\text{max}} = 2$ with varying r_{cut} are shown in fig. 3.24. The history for $l_{\text{max}} = 0$, which corresponds to scalar features only, is not shown due to rapid convergence to high error levels. Owing to the cosine annealing schedule, convergence required extended training times. Training was stopped once a reasonable plateau was reached. Depending on the random initialization and dataset partitioning, training results varied by up to 10% in terms of test-error metrics.

The resulting test errors are summarized in tab. 3.7. In general, accuracy improves with increasing cut-off radius, though the largest cut-off did not provide significant gains compared to $r_{\text{cut}} = 3.0$. The scalar model, $l_{\text{max}} = 0$, underperforms even relative to n2p2 and DeePMD, showing the importance of higher-order features. Including first-order contributions with $l_{\text{max}} = 1$ already yields superior performance to the other architectures, while the addition of second-order tensorial features, $l_{\text{max}} = 2$, provides even further improvement, albeit at significantly higher computational cost. Including even higher tensorial orders was not attempted due to prohibitive resource requirements, demanding more memory than the 48 GB provided by an RTX A6000 GPU.

Overall, NequIP with $l_{\text{max}} = 1$ and $l_{\text{max}} = 2$ substantially outperformed both n2p2 and DeePMD. Based on these results, the preferedd potential for subsequent simulations

[‡]Exponential decay was not yet available in the standard package at the time.

consists of rotational orders $l_{\max} = 1$ or $l_{\max} = 2$ with cut-off radius $r_{\text{cut}} = 3.0$, the choice of the former being determined by the tradeoff between accuracy and efficiency. Notably, NequIP was able to achieve good performance even when trained exclusively on forces, yielding force accuracies comparable to combined force and energy training. However, training without forces fails. The resulting force errors are far too large to permit stable molecular dynamics simulations. This underlines the necessity of including force information in training for the Voronoi potential.

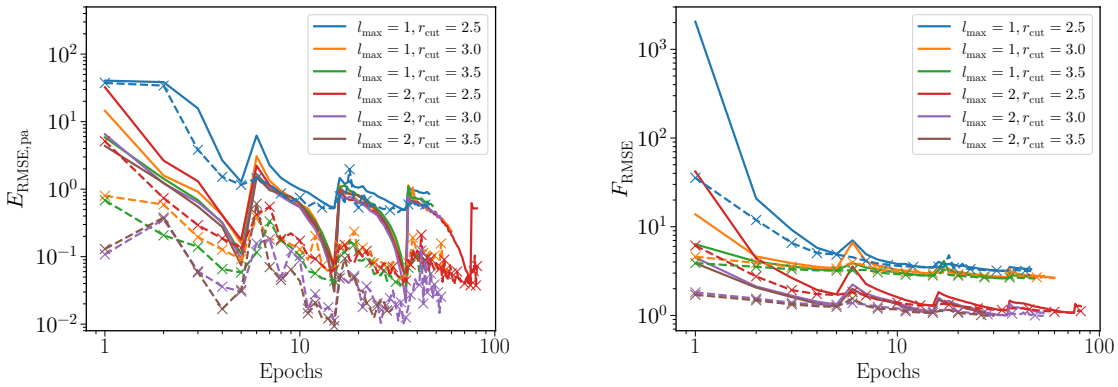


Figure 3.24: Training history of Deep Potentials for the Voronoi potential using NequIP with tensorial features up to $l_{\max} = 1$ and $l_{\max} = 2$ and different cutoff radii r_{cut}. The plots show the energy errors (left) and force errors (right) as functions of the training epoch. Solid lines represent training errors, while dashed lines denote validation errors. Training errors are computed as running averages within each epoch.

$l_{\max}$	r_{cut}	$\mathcal{W}_{\text{E}}$	$\mathcal{W}_{\text{F}}$	$E_{\text{RMSE,pa}}$	F_{RMSE}	$F_{\text{RMSE}}/F_{\text{mean}}$
0	2.5	1	10	5.51	11.5	0.145
0	3.0	1	10	0.937	9.71	0.123
0	3.5	1	10	0.763	9.91	0.125
1	2.5	1	10	0.455	3.18	0.0402
1	3.0	1	10	0.0468	2.63	0.0333
1	3.5	1	10	0.0366	2.60	0.0329
2	2.5	1	10	0.0372	1.08	0.0137
2	3.0	1	10	0.0103	0.985	0.0125
2	3.5	1	10	0.00934	0.985	0.0125
2	3.0	0	1	54.4	1.01	0.0128
2	3.0	1	0	0.130	17.2	0.217

Table 3.7: Comparison of NequIP models for the Voronoi potential with varying cutoff radii r_{cut}, maximum rotational orders $l_{\max}$, and relative error weights $\mathcal{W}_{\text{E}}$ and $\mathcal{W}_{\text{F}}$. The table reports the resulting energy and force errors on the test set, along with the relative force error with respect to the mean force magnitude.

Summary

Among the investigated architectures, NequIP consistently outperforms both n2p2 and DeePMD, which showed comparatively large force errors. Whether such errors are acceptable can ultimately only be judged in actual simulations, where the stability of trajectories and the ability to reproduce structural and dynamical observables provide a more stringent test than simple error metrics. The best-performing potential should balance accuracy with computational efficiency, though NequIP clearly offers the most reliable architecture for the Voronoi system.

Having established that the Voronoi potential can be represented by high-quality NNPs, the focus now shifts from training error metrics to the properties of the resulting models, such as long-term stability, conservation laws, and the faithful reproduction of slow relaxation processes.

3.3.3 Potential Evaluation

While the general quality of NNPs is assessed by monitoring global error metrics during training, a more thorough evaluation is required to determine their reliability. To this end, additional test sets are generated which are independent from the training and validation data, following the same methodology outlined in sec. 3.3.1. This ensures a large overlap for the relevant configuration space while avoiding any explicit or implicit bias from configurations used during optimization.

Three networks are considered. The k_{NN}-DeePMD implementation with a neighbor cut-off of $k_{\mathrm{NN}} = 128$, as well as two NequIP models with cut-off radius $r_{\mathrm{cut}} = 3.0$ and maximum rotational orders $l_{\mathrm{max}} = 1$ and $l_{\mathrm{max}} = 2$, respectively.

Scatter plots of reference versus predicted energies and forces provide a visualization of the predictive performance, see fig. 3.25. Configurations with $\rho = 1.0$ and temperatures $T \in [0.25, 50]$ are evaluated. Energies are generally reproduced well across all networks. However, forces reveal a more differentiated picture, the k_{NN} Deep Potential shows a pronounced spread, with predicted forces systematically underestimated in magnitude. In contrast, the NequIP models exhibit far better agreement, with $l_{\mathrm{max}} = 2$ showing the smallest variance in force deviations, consistent with its lower total error metrics obtained during training.

A complementary view is provided in fig. 3.26, where differences in force magnitudes and relative angles between predicted and reference forces are shown. DeePMD exhibits the broadest spread, followed by NequIP with $l_{\mathrm{max}} = 1$. NequIP with $l_{\mathrm{max}} = 2$ demonstrates a remarkably narrow distribution of both force magnitude differences and angular deviations.

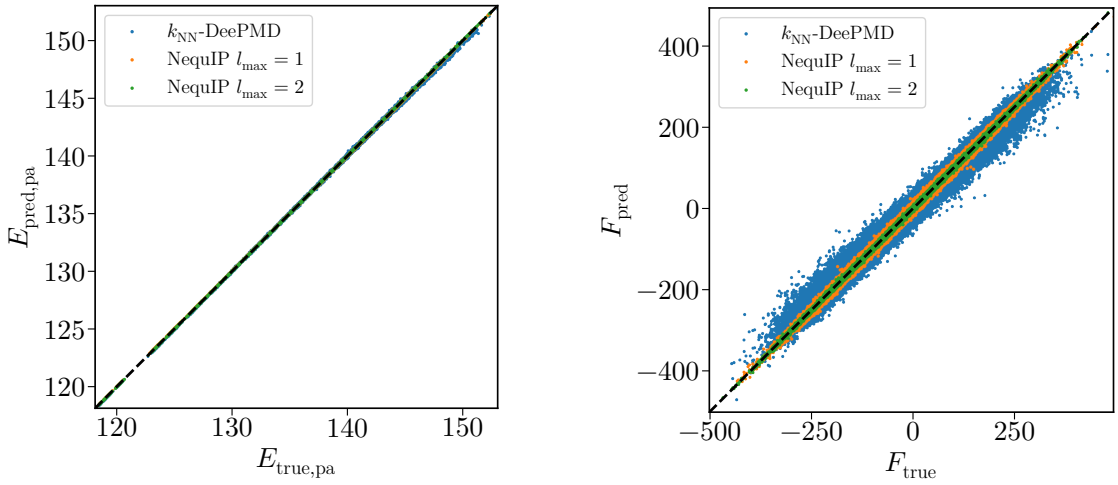


Figure 3.25: Error scatter plots comparing reference values of the Voronoi system with predictions from different neural network potentials for configurations at density $\rho = 1.0$ and temperatures $T \in [0.25, 50]$. The panels display total energies per atom (left) and force components (right). Diagonal lines represent perfect agreement.

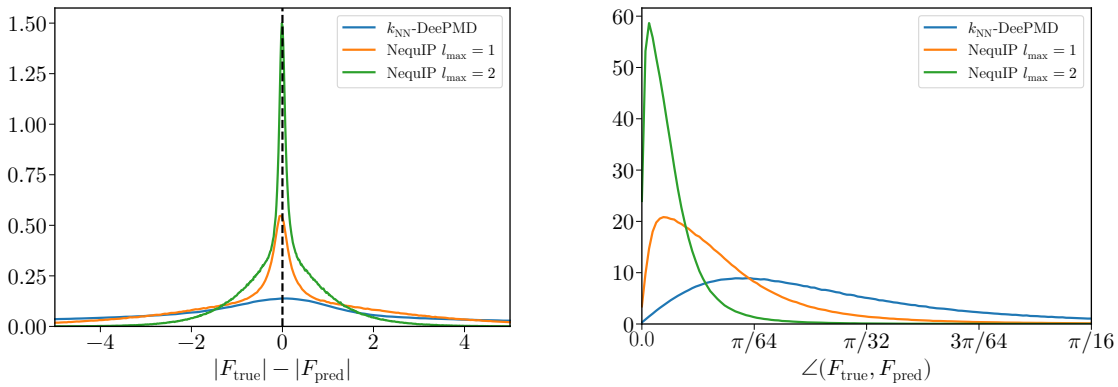


Figure 3.26: Histogram of force magnitude differences (left) and angular deviations (right) between reference forces of the Voronoi system and predictions from different neural network potentials for configurations at density $\rho = 1.0$ and temperatures $T \in [0.25, 50]$.

In contrast to typical DFT-based potentials, the Voronoi interaction allows a natural decomposition of the total energy into per-particle contributions. While these atomic energies are formally well-defined in the Voronoi model, their physical interpretability in the context of NNPs should be taken with care. Fig. 3.27 presents scatter plots of predicted versus reference atomic energies. As expected, no meaningful agreement is found, since the networks were not explicitly trained on these quantities. In particular, NequIP predictions appear almost orthogonal to the expected trend, reflecting the substantial redistribution of energy contributions. This highlights the auxiliary role of atomic energies in NNPs, as they serve as variables to reconstruct the correct PES, but need not coincide with the underlying atomic energies. Including atomic energies in the training optimization was found to be detrimental, as even including a small weight on atomic-energy errors prevented convergence or at least degraded performance significantly.

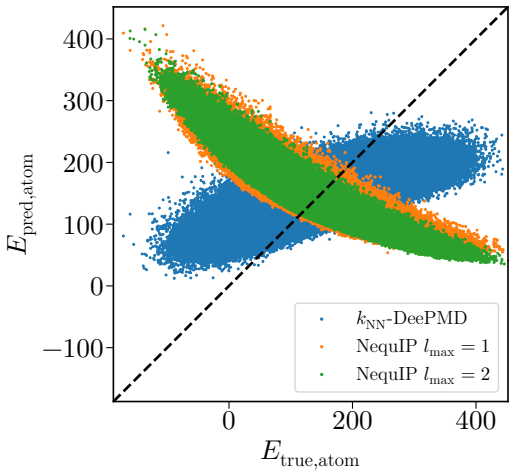


Figure 3.27: Error scatter plot comparing reference atomic energies of the Voronoi system with predictions from different neural network potentials for configurations at density $\rho = 1.0$ and temperatures $T \in [0.25, 50]$. Diagonal lines represent perfect agreement.

Although the Voronoi interaction does not permit the definition of two-particle energies and forces, it is interesting to evaluate such isolated configurations with the trained NNPs. To contruct a finite Voronoi cell, at least four neighbor particles are required in three dimensions. Tesselations for pairs, triplets, and quadruplets are not possible. Nevertheless, the extrapolation to such states can remain stable for well-converged networks, given that it operates in its approximately linear regime.

In fig. 3.28, the energies and forces of a two-particle configuration as a function of distance, as predicted by NequIP with $l_{\mathrm{max}} = 2$ and $r_{\mathrm{cut}} = 3.0$, are shown. For comparison, a typical radial distribution function of the Voronoi liquid, scaled and shifted, is included. Interestingly, the first peak of the distribution roughly aligns with the maximum of the predicted potential energy. This suggests that the neural network has effectively learned

a quantity resembling the potential of mean force ($E \approx -k_\mathrm{B}T \ln g(r)$) [44], albeit with different sign and magnitude. The effective interaction is not repulsive at small distances and only exhibits repulsive behavior close to the first structural peak. This is consistent with the stabilizing mechanism of the Voronoi potential, where stability emerges from the geometry of cells rather than from explicit repulsion. Analysis of triplets confirms this picture, where the central particle typically experiences forces that either pull it towards the other particles or push it away, depending on interparticle distances.

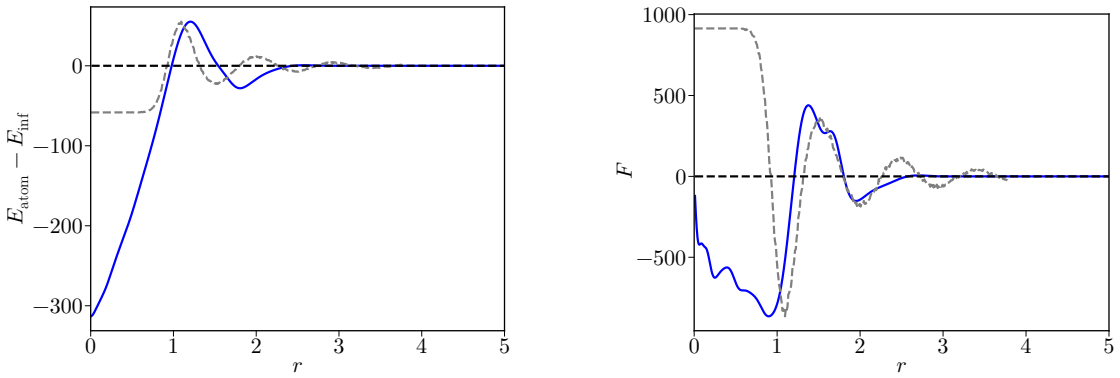


Figure 3.28: Energies (left) and forces (right) for a pair of particles at relative distance r, as predicted by the trained NequIP model with $l = 2$ for the Voronoi potential. One particle is fixed on the x-axis, and the energies and forces are evaluated for a second particle moving along the positive x-direction. For comparison, the gray dashed lines show scaled and shifted radial distribution functions of a representative Voronoi liquid, adjusted to match the magnitudes of the energies and forces.

The quality of the potentials can also be assessed through energy-conservation tests in the microcanonical ensemble, as outlined in sec. 3.2.2. For $T = 1.0$ and $\rho = 1.0$, the fluctuations of the total energy obtained from simulations with NequIP are compared with those of the reference Voronoi potential in fig. 3.29. The agreement is excellent, with both models exhibiting essentially identical fluctuation amplitudes. Although their mean absolute energies differ by more than the fluctuation scale, this is inconsequential, as constant energy offsets are physically irrelevant and only energy differences and forces matter. The standard deviation of total-energy fluctuations is shown in fig. 3.30, and both the reference and the NequIP model display the expected $\propto \Delta t^2$ scaling of the error.

An interesting phenomenon is observed when performing simulations with the k_NN-DeePMD model at high temperatures, near the upper boundary of the training range. In this regime, some systems collapse rapidly into multiple compact clusters during equilibration, as illustrated in fig. 3.31. These clusters, containing on average $\langle N \rangle = 20.48 \pm 0.91$ particles with a radius of $\langle r \rangle = 0.365 \pm 0.011$, are remarkably stable and remain uniform in size and composition throughout the simulation. Cluster identification was performed using a

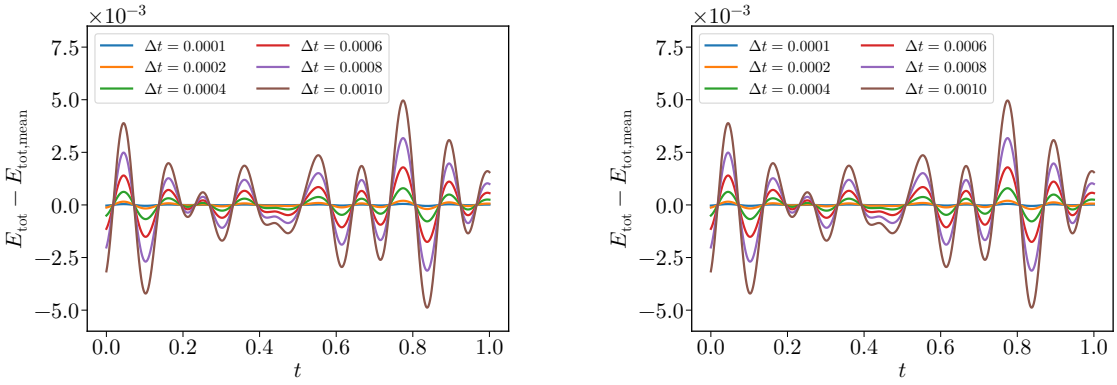


Figure 3.29: Standard deviation of microcanonical energy fluctuations for the Voronoi system (left) and the NequIP model with $l_{\text{max}} = 2$ (right), evaluated at $T = 1.0$ and $\rho = 1.0$ for different discretization time-steps Δt.

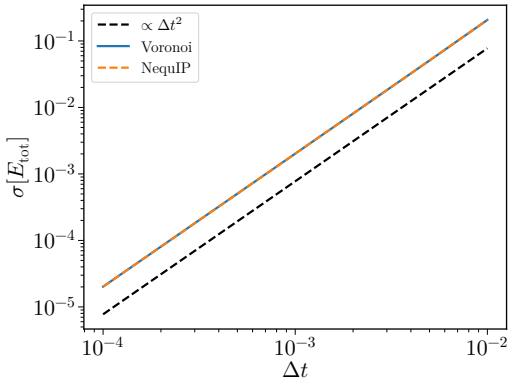


Figure 3.30: Standard deviation of microcanonical energy fluctuations as a function of the discretization time-step Δt for the Voronoi system and the NequIP model with $l_{\text{max}} = 2$ (dashed), evaluated at $T = 1.0$ and $\rho = 1.0$.

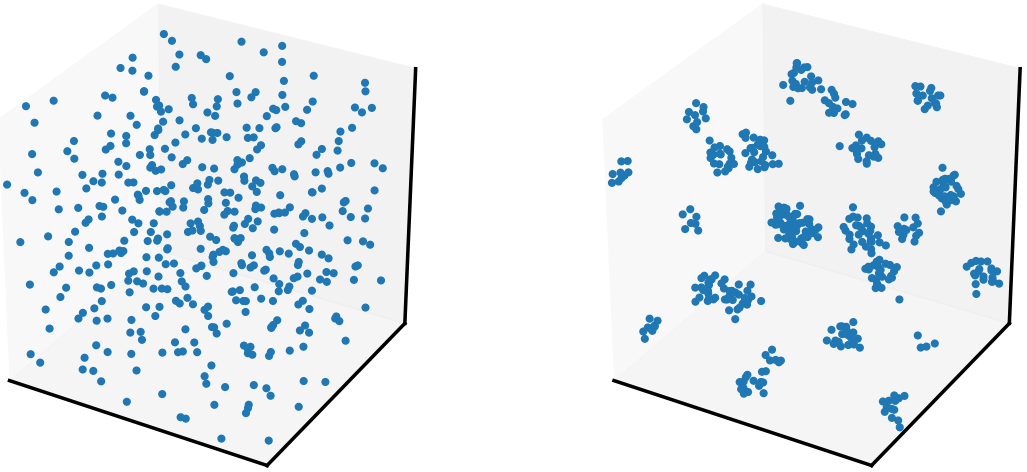


Figure 3.31: Snapshots of a simulation using the k_{NN}-DeePMD model for the Voronoi potential at $T = 25$, $\rho = 1.0$. The left panel shows the system shortly before collapse, while the right panel depicts the configuration after the collapse.

distance-based connectivity threshold of 0.5 and averaged over 100 configurations.

All NequIP simulations remained stable, highlighting the limitations of the DeePMD model. The observed collapse could be attributed to the large spread of force errors and the poor generalization capacity of the k_{NN}-based descriptor at higher temperatures. The underlying Voronoi potential is stabilized by edge effects. In the absence of boundaries, the minimal-energy state would correspond to all particles collapsing into a single point, which is avoided by large Voronoi cells forming at the edges. The NNP, however, lacks this stabilizing mechanism, and once clusters form, forces are too small to break particles free.

In general, reducing force errors is expected to improve the dynamics in simulations with respect to the reference system. However, no a priori definition of a "reasonable" force error can be given. Whether a NNP is sufficient for the given task should be judged based on simulation results. Since many force components in the Voronoi system are close to zero, rare outliers can dominate the error distribution but may be insignificant to the overall dynamics in simulations.

In the next section, NNPs are evaluated based on their performance in MD simulations, where small inaccuracies in the potential-energy surface may propagate into dramatic deviations in the dynamics.

3.3.4 Simulation Comparison

To properly assess the quality of the neural network potentials, their performance is evaluated in MD simulations, comparing structural and dynamical observables between the reference Voronoi potential and the trained NNPs. In this context, stability issues may emerge that remain invisible when analyzing static configurations, particularly at higher temperatures where systems possibly explore configurations far from those sampled in the training set.

All simulations are performed using the same protocol, with the only difference being the choice of potential. Ensembles of 500 particles are generated for each state point, starting from equilibrated high-temperature ($T = 10$) configurations at density $\rho = 1.0$. The NNPs were trained on systems of only 250 particles. Due to limitations in the NequIP implementation at the time of this work (absence of MPI support), simulations with more than a few hundred to a thousand particles were not feasible. Within each production run, sub-ensembles are sampled to improve statistics under the assumption of ergodicity. Equilibration runs of 10^5 steps with a time-step $\Delta t = 10^{-3}$ are carried out in the NVT ensemble, followed by production runs of 10^6 steps in the NVE ensemble.

In terms of computational performance, simulations with the reference Voronoi potential take roughly 8 h on a single AMD EPYC 7702 core. Parallelization of the tessellation using Voro++ becomes efficient only for system sizes of a few thousand particles, with nearly linear scaling once each processor handles about a 1000 ghost particles. The custom k_{NN}-DeePMD implementation requires approximately 12 h on a single NVIDIA RTX A6000 GPU, utilizing about 50% of the hardware due to inefficient interfacing between PyTorch and LAMMPS. In particular, the network needs to calculate neighbor lists internally via PyTorch3D instead of reusing the optimized LAMMPS neighbor lists, which leads to a substantial performance bottleneck. The official LAMMPS interface of DeePMD avoids these issues but is restricted to radial cut-offs. NequIP simulations are significantly more expensive, where a model with $l_{\mathrm{max}} = 1$ requires about 50 h on an RTX A6000, while increasing the tensorial order to $l_{\mathrm{max}} = 2$ extends this to roughly 120 h. As expected, larger and more complex NNPs demand substantially more resources[‡], with energy and force evaluations dominating the wall time for all simulations.

Simulation results are shown in fig. 3.32. The phase behavior, including the emergence of meta-stable liquid states, is analyzed in [22]. Stable crystalline phases are predicted for $T \lesssim 1.65 - 1.85$, with a bcc crystal being the most stable structure, whereas meta-stable fluids occur for $T \gtrsim 1.05$. Radial distribution functions and self-intermediate scattering

[‡]Recent versions of PyTorch and NequIP can improve performance by up to an order of magnitude, as suggested by small-scale tests. The performance values reported here were obtained with an older version (current at the time of the study) and are retained for completeness.

functions show good agreement between the reference Voronoi system and the trained potentials. As anticipated from the training results, NequIP substantially outperforms DeePMD. The latter fails to reproduce the meta-stable liquid state at $T = 1.25$ despite multiple attempts, confirming an inherent instability in the DeePMD model (compare fig. 3.31). Other meta-stable states are also captured less reliably compared to both NequIP models. Although relatively poor force predictions of DeePMD are consistent with its training errors, this deficiency has only limited impact on the dynamics of stable solids and liquids. Reproducing meta-stable states is challenging given the limited number of independent ensembles available and the long relaxation times. Deviations are not unexpected and may also reflect effects of finite sampling rather than intrinsic flaws of the potential.

Stable liquids and crystalline phases are well reproduced by all NNPs, with crystallization into the expected bcc-structure. NequIP with $l_{\max} = 1$ and $l_{\max} = 2$ both provide stable dynamics across all tested state-points, despite a nearly threefold difference in force errors. This indicates that once a sufficiently expressive architecture is used, the exact force accuracy becomes less critical for stability. At high temperatures, $T > 10$, dynamics become trivial and are not shown here. In this regime, the DeePMD model also displays its numerical instabilities.

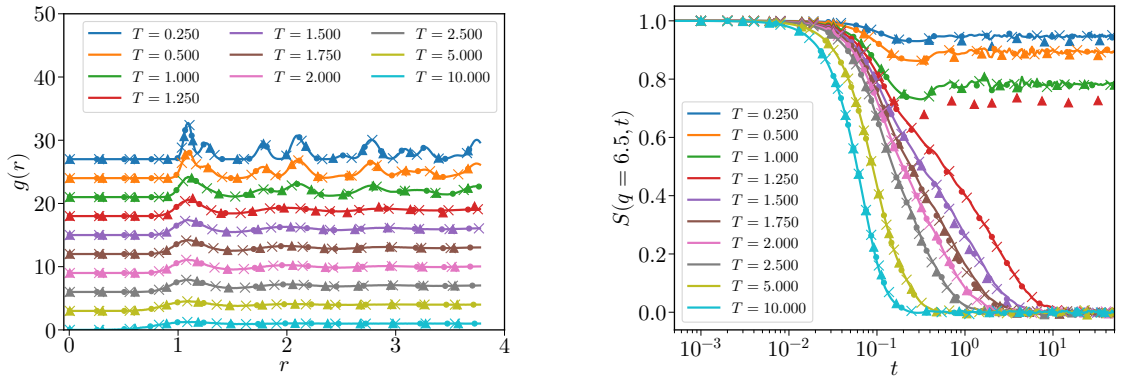


Figure 3.32: Comparison of radial distribution functions (RDFs, left) and self-intermediate scattering functions (SISFs, right) between simulations with the Voronoi potential (solid) and the trained neural network potentials k_{NN}-DeePMD (triangles), NequIP with $l_{\max} = 1$ (dots), and NequIP with $l_{\max} = 2$ (crosses), at density $\rho = 1.0$ for different temperatures T.

Summary

Several NNPs have been trained and evaluated for the single-component Voronoi potential. Despite considerable differences in training performance, with NequIP clearly outperforming the other approaches, all investigated models proved adequate in reproducing the underlying system, at least within a restricted region of phase-space. Through careful dataset preparation and a consistent training methodology, reliable potentials and accurate simulations could be obtained.

While meta-stable states exhibit slow dynamics, eventual crystallization complicates the investigation of glass-forming behavior. To explore the capability of NNPs in the context of glassy dynamics, a binary variant of the Voronoi potential is proposed. With the experience gained regarding dataset design and training procedures, the NequIP architecture will be employed to model this system. Owing to its design already incorporating particle species in the input, no modifications are required. The following section thus turns to NNPs for this system.

3.4 Binary Voronoi System

To overcome the tendency of the single-component Voronoi fluid to crystallize and enable the study of NNPs for glassy physics and increasingly slow dynamics, a polydisperse variant of the model is introduced. Polydispersity is realized by assigning different effective radii R_i to the particles and constructing a radical Voronoi tessellation, in which the cell boundaries are weighted by the relative particle sizes [82].

The potential energy and forces are given by [23]

$$E_{\mathrm{pot}} = \sum_{i=1}^{N} E_{\mathrm{pot},i} = \frac{\gamma}{2} \sum_{i=1}^{N} \int_{v_i} \mathrm{d}^3\boldsymbol{r} \ \left(r^2 - R_i^2 + R^2\right), \tag{3.23}$$

$$\boldsymbol{F}_i = \gamma \boldsymbol{\tau}_i = \gamma \int_{v_i} \mathrm{d}^3\boldsymbol{r} \ \boldsymbol{r} = \gamma v_i \left(\boldsymbol{r}_i - \boldsymbol{r}_{\mathrm{c},i}\right), \tag{3.24}$$

where $R^2 = N^{-1} \sum_i R_i^2$ denotes the mean squared radius of particles. This additional R^2 term ensures that the monodisperse limit remains consistent with the original formulation, see eq. (3.22). The actual difference in energies and forces arises from the modified Voronoi cells with volume v_i.

To avoid the complexity of fully polydisperse systems, where each distinct radius would constitute a separate species, a binary mixture is considered. Particles of type A are assigned radius $R_{\mathrm{A}} = 0.95$ and particles of type B radius $R_{\mathrm{B}} = 1.05$, with the composition of the system specified by the concentration of A particles χ_{A} and $\chi_{\mathrm{B}} = 1 - \chi_{\mathrm{A}}$. This system avoids crystallization and exhibits glassy behavior upon cooling, making it ideal for probing the performance of NNPs in this context.

Care must be taken to choose moderate values of polydispersity to maintain physically stable tessellations. An excessive size disparity may cause small particles to lie outside their assigned Voronoi cells or even lead to vanishing cell volumes [88]. Similarly, temperatures should remain below a threshold beyond which the geometry of the Voronoi cells becomes unstable [23].

Analogous to the single-component case, a NNP will be trained for this binary system, allowing for detailed comparisons between the structural and dynamical properties obtained from simulations with the exact Voronoi potential and those generated with the trained models.

3.4.1 Network Training

For the binary Voronoi system, the NequIP architecture is employed unless otherwise noted, as it outperformed both DeePMD and n2p2 for the single-component system in terms of

training accuracy and stability. A major advantage of NequIP is its natural incorporation of particle species as direct input features on the graph nodes.

Since the binary system does not fundamentally alter the PES compared to the single-component fluid, the training data is sampled from a similar region of phase-space. The dataset consists of 50 000 configurations, each containing 500 atoms. Compared to the single-component setup, the number of particles was doubled to ensure sufficient representation of local environments for both particle species.

Training data was generated as for the single-component case, see sec. 3.3.1. Simulations of the Voronoi potential, including the modified tessellation with distinct particle radii, are initialized from randomized states, equilibrated in the NVT ensemble, and subsequently sampled in the NVE ensemble at regular intervals to form the dataset. The phase-space range was chosen with temperatures $T \in [0.25, 50.0]$, densities $\rho \in [0.8, 1.25]$, and particle concentrations $\chi_\mathrm{A} \in [0.0, 1.0]$, covering both monodisperse limits as well as the mixed system.

Including both monodisperse limits in the dataset is essential, as they anchor the training data at the boundaries of pure A and pure B systems. In these cases, the radical Voronoi tessellation reduces to the standard tessellation, and the particle interactions coincide with those of the single-component fluid. Omitting the monodisperse limits severely reduces stability and generalization capability of the network.

For the network architecture, a cut-off radius of $r_\mathrm{cut} = 3.0$ is selected and models with tensorial orders $l_\mathrm{max} = 1$ and $l_\mathrm{max} = 2$ are trained using the same hyperparameters as determined in sec. 3.3.2. Training converged efficiently, yielding energy and force errors comparable to the single-component case, although the radical tessellation leads to larger average force magnitudes. The training histories are shown in fig. 3.33, while the final error metrics evaluated on a testset are summarized in tab. 3.8, including a comparison with n2p2 using nearly 300 symmetry functions.

As expected, NequIP outperforms n2p2, and higher rotational orders provide better accuracy. This reconfirms that the equivariant graph architecture is particularly well suited to capture the complex interactions of the Voronoi interaction.

		$E_\mathrm{RMSE,pa}$	F_RMSE
NequIP	$l_\mathrm{max} = 1$	0.154	2.98
NequIP	$l_\mathrm{max} = 2$	0.059	1.07
n2p2	$N_\mathrm{SF} = 296$	0.074	6.58

Table 3.8: Comparison of error metrics for trained neural network potentials of the binary Voronoi system with a cutoff radius of $r_\mathrm{cut} = 3.0$, evaluated on a test set.

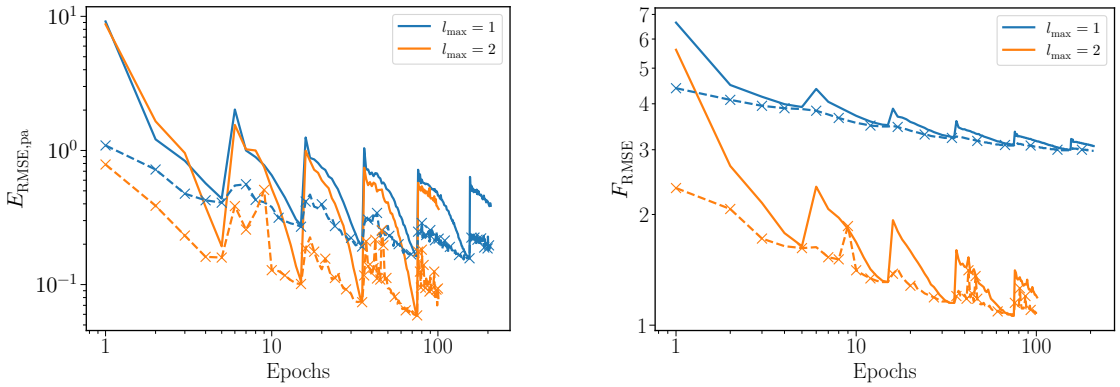


Figure 3.33: Training history of NequIP models for the binary Voronoi potential using tensorial features of order $l_{\mathrm{max}} = 1$ and $l_{\mathrm{max}} = 2$ with a cutoff radius of $r_{\mathrm{cut}} = 3.0$. Solid lines represent training errors, while dashed lines with crosses indicate validation errors. Training errors are computed as running averages within each epoch. The left panel shows the total energy error per atom, and the right panel shows the error of the force components.

3.4.2 Potential Evaluation

The analysis of the NNPs is restricted to the two NequIP models with $l_{\mathrm{max}} = 1$ and $l_{\mathrm{max}} = 2$. Configurations for a binary mixture with $\chi_{\mathrm{A}} = \chi_{\mathrm{B}} = 0.5$ as well as configurations in the monodisperse limits $\chi_{\mathrm{A}} = 1.0$ and $\chi_{\mathrm{B}} = 1.0$ are evaluated with the NNPs. The HDNNP was not included due to its prohibitive computational cost, rendering it impractical for simulations with the current implementation of n2p2. In reality, the symmetry function approach remains far more efficient than constructing a graph neural network with multiple layers, even for several hundred or even thousand symmetry functions, at least for just two different species.

For the binary mixture with the same concentration of A and B particles, the NequIP models reproduce total energies and force components with high accuracy for both $l_{\mathrm{max}} = 1$ and $l_{\mathrm{max}} = 2$, see fig. 3.34. As already observed in the single-component case, see sec. 3.3.3, atomic energy predictions prove physically meaningless and show regression lines nearly perpendicular to the expected values. Two distinct lines for A and B particles are visible, reflecting the larger average energy of the B particles due to their on average larger Voronoi cells. Force prediction deviations follow a similar pattern, see fig. 3.35. Interestingly, force magnitudes for A particles are reproduced slightly better, while force directions are more accurate for B particles, possibly due to the magnitude of the force components in the loss function.

In the monodisperse limits, an asymmetry between the A and B systems is observed. While the force errors remain nearly identical, energy errors are marginally higher for the

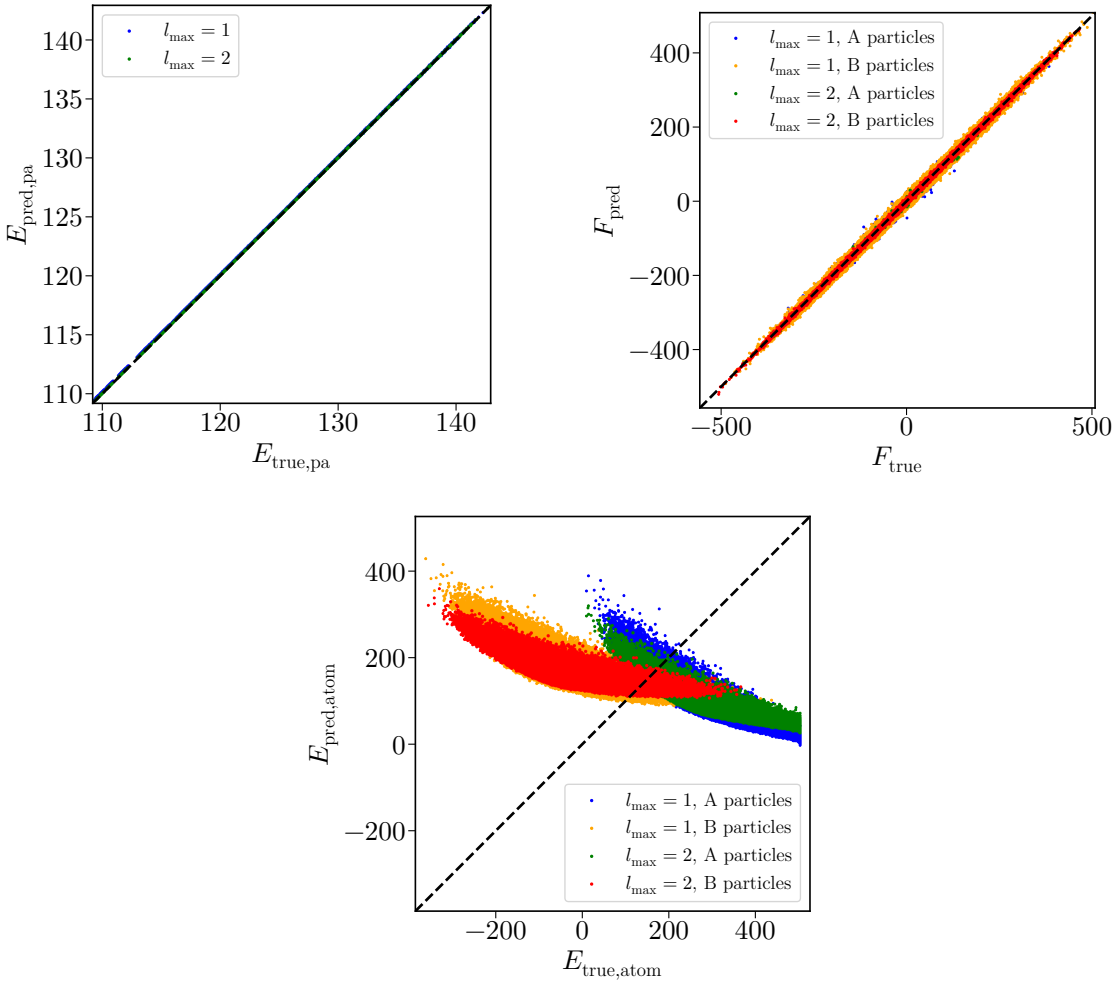


Figure 3.34: Error scatter plots comparing reference values of the binary Voronoi system with predictions from NequIP models using tensorial orders $l_{\max} = 1$ and $l_{\max} = 2$ for configurations at density $\rho = 1.0$, temperatures $T \in [0.25, 50]$, and composition $\chi_A = \chi_B = 0.5$. The panels display total energies per atom (top left), force components (top right), and atomic energies (bottom). Diagonal lines represent perfect agreement.

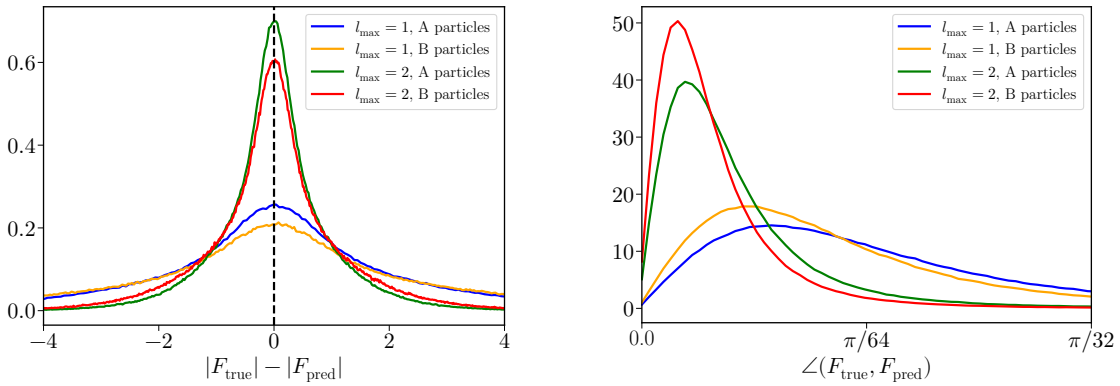


Figure 3.35: Histogram of force magnitude differences (left) and angular deviations (right) between reference forces of the binary Voronoi system and predictions from the NequIP model with tensorial orders $l_{max} = 1$ and $l_{max} = 2$ for configurations at density $\rho = 1.0$, temperatures $T \in [0.25, 50]$, and composition $\chi_A = \chi_B = 0.5$.

pure A system ($E_{RMSE,pa} = 0.081571$, $F_{RMSE} = 1.0223$) compared to the pure B system ($E_{RMSE,pa} = 0.033686$, $F_{RMSE} = 1.0158$). The scatter plot in fig. 3.36 reveals distinct regression lines for atomic energies, confirming different redistributions of the total energy in the monodisperse limits. Force magnitude and direction errors, see fig. 3.37, indicate the differences between pure A and B systems for these NNPs. This likely originates from the training on mixed configurations, where the larger B particles dominate the optimization compared to pure systems.

Energy conservation tests in the NVE ensemble yield results similar to the single-component system for mixed systems and the monodisperse limits and are therefore not shown. It should be noted, however, that the underlying radical Voronoi potential itself exhibits stability issues for very high temperatures or excessive polydispersity, where particles may end up outside their cells. This is actively avoided here by limited polydispersity and probed temperature range.

Overall, these results confirm that the trained NequIP models can reproduce the binary Voronoi system, but also highlight the subtle asymmetries for pure A and B particle interactions, while the underlying system shows the same tesselation for both systems. Such deviations are unlikely to fundamentally affect the dynamics of the pure systems, but could become problematic for mixtures with increasing χ_A. Assessing such effects requires simulations of structural and dynamical observables, which will be presented in the following section.

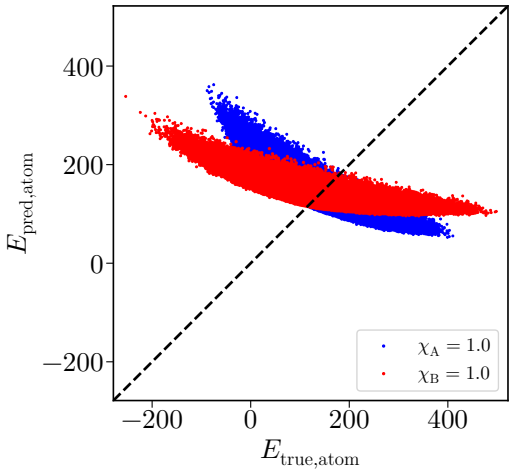


Figure 3.36: Error scatter plot comparing reference atomic energies of the binary Voronoi system with predictions from the NequIP model using tensorial order $l_{\text{max}} = 2$ for configurations at density $\rho = 1.0$ and temperatures $T \in [0.25, 50]$ in the monodisperse limits $\chi_A = 1.0$ and $\chi_B = 1.0$. A diagonal line represents perfect agreement.

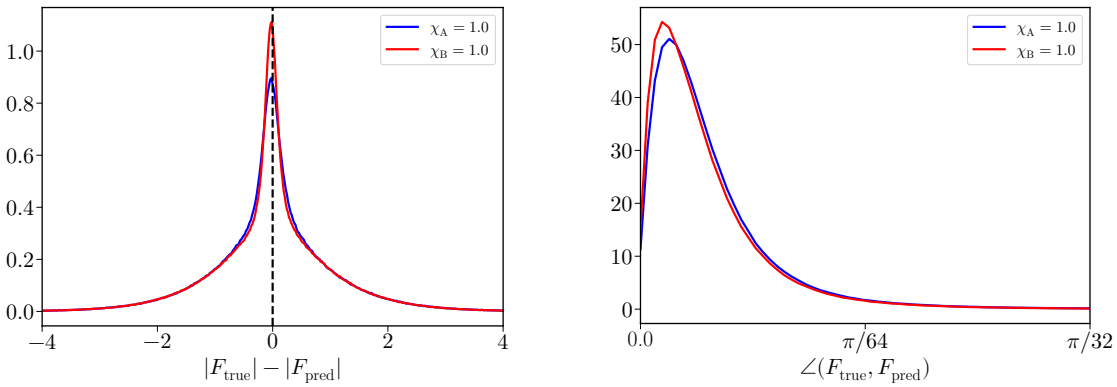


Figure 3.37: Histogram of force magnitude differences (left) and angular deviations (right) between reference forces of the binary Voronoi system and predictions from the NequIP model with tensorial order $l_{\text{max}} = 2$ for configurations at density $\rho = 1.0$ and temperatures $T \in [0.25, 50]$ in the monodisperse limits $\chi_A = 1.0$ and $\chi_B = 1.0$.

3.4.3 Simulation Comparison

To assess the quality of the NNPs beyond static error metrics, MD simulations are conducted. For each state point, a single ensemble of 500 particles is simulated, initialized from randomized configurations with small overlap. To enhance statistical accuracy, averages are taken over sub-ensembles extracted from the trajectories, assuming ergodicity, which is justified as crystallization is suppressed by the polydispersity. Simulations are restricted to $\rho = 1.0$, since density merely introduces scaling of lengths and energies. Different compositions are highlighted, specifically $\chi_\mathrm{A} = 0.2$, $\chi_\mathrm{A} = 0.5$, and $\chi_\mathrm{A} = 0.8$. All simulated systems remained stable and free of crystallization. Systems with more A particles show longer relaxation times than those consisting of more B particles.

Radial Distribution Functions

The structural properties are evaluated via radial distribution functions (RDFs), shown in figs. 3.38 to 3.40 for $\chi_\mathrm{A} = 0.2$, 0.5, 0.8, respectively. Static structure factors do not provide further insight and are omitted. Across all concentrations, the RDFs demonstrate good agreement between the NNP and the reference Voronoi simulations. The best agreement is observed for mixtures with equal concentrations and for systems dominated by A particles. In contrast, the $\chi_\mathrm{A} = 0.2$ system exhibits more noticeable deviations, particularly at higher temperatures and in the AA correlations. This behavior is consistent with the reduced statistical weight of the smaller, less numerous A particles, which may also reflect the asymmetry between A and B interactions for the NNPs. B particles contribute more to the total energy and forces owing to their larger size. Nevertheless, the static structure of the binary systems is overall reproduced pretty well.

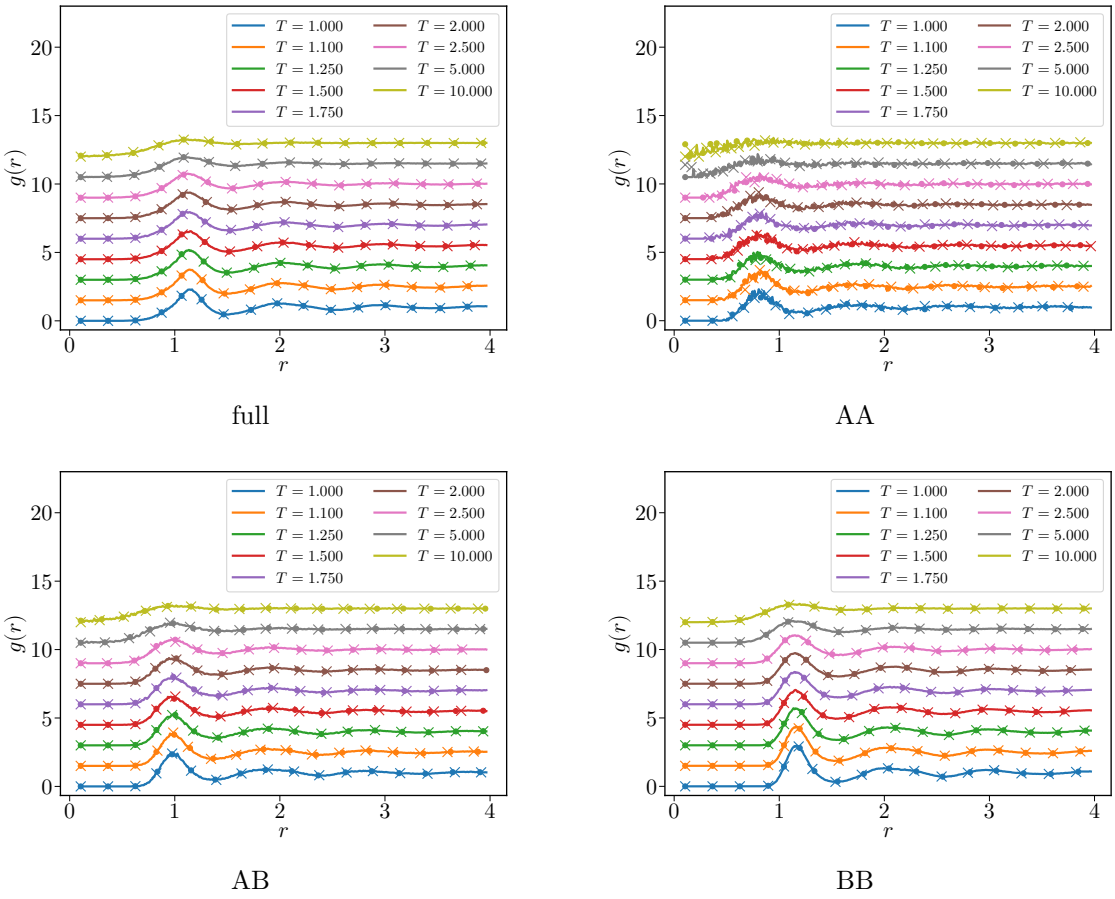


Figure 3.38: Radial distribution functions of the binary Voronoi fluid (solid lines) compared with simulations using NequIP with $l_{\text{max}} = 1$ (crosses) and $l_{\text{max}} = 2$ (dots) at density $\rho = 1.0$, concentration $\chi_{\text{A}} = 0.2$, and various temperatures T.

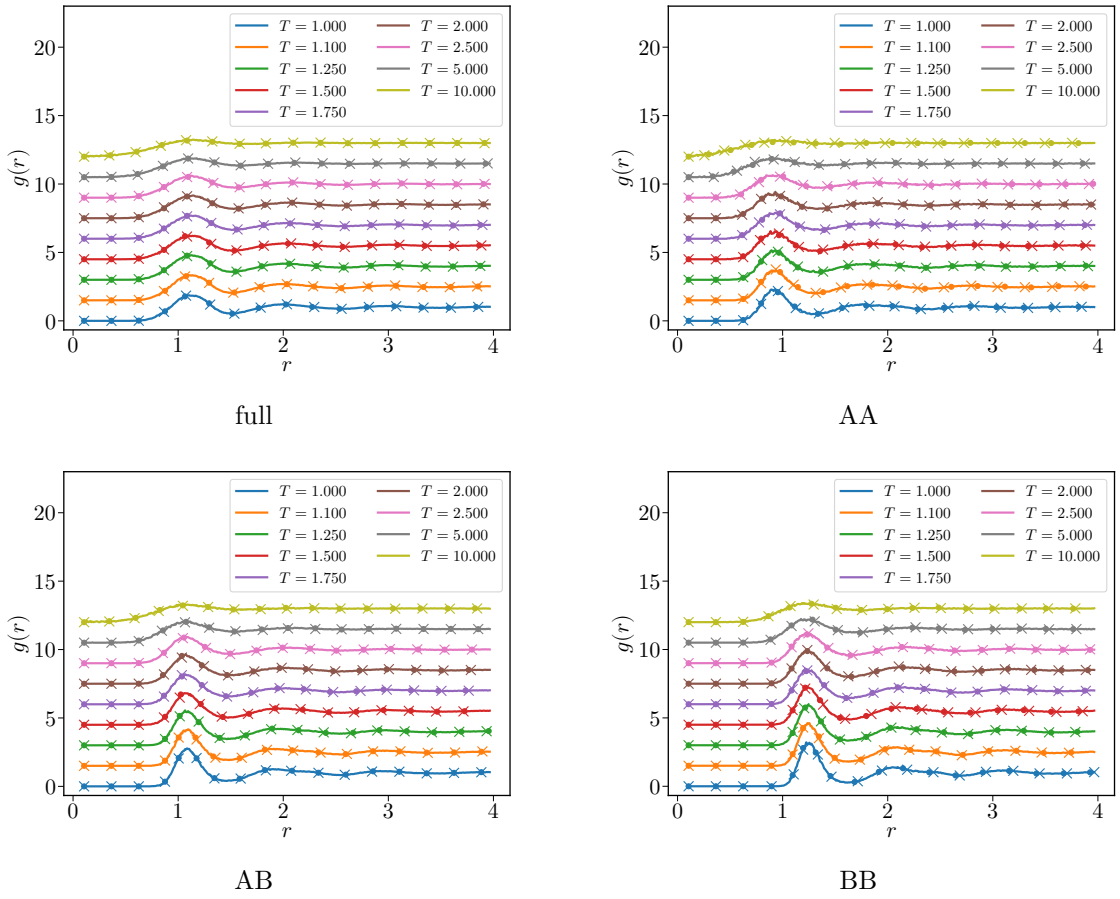


Figure 3.39: Radial distribution functions of the binary Voronoi fluid (solid lines) compared with simulations using NequIP with $l_{\text{max}} = 1$ (crosses) and $l_{\text{max}} = 2$ (dots) at density $\rho = 1.0$, concentration $\chi_{\text{A}} = 0.5$, and various temperatures T.

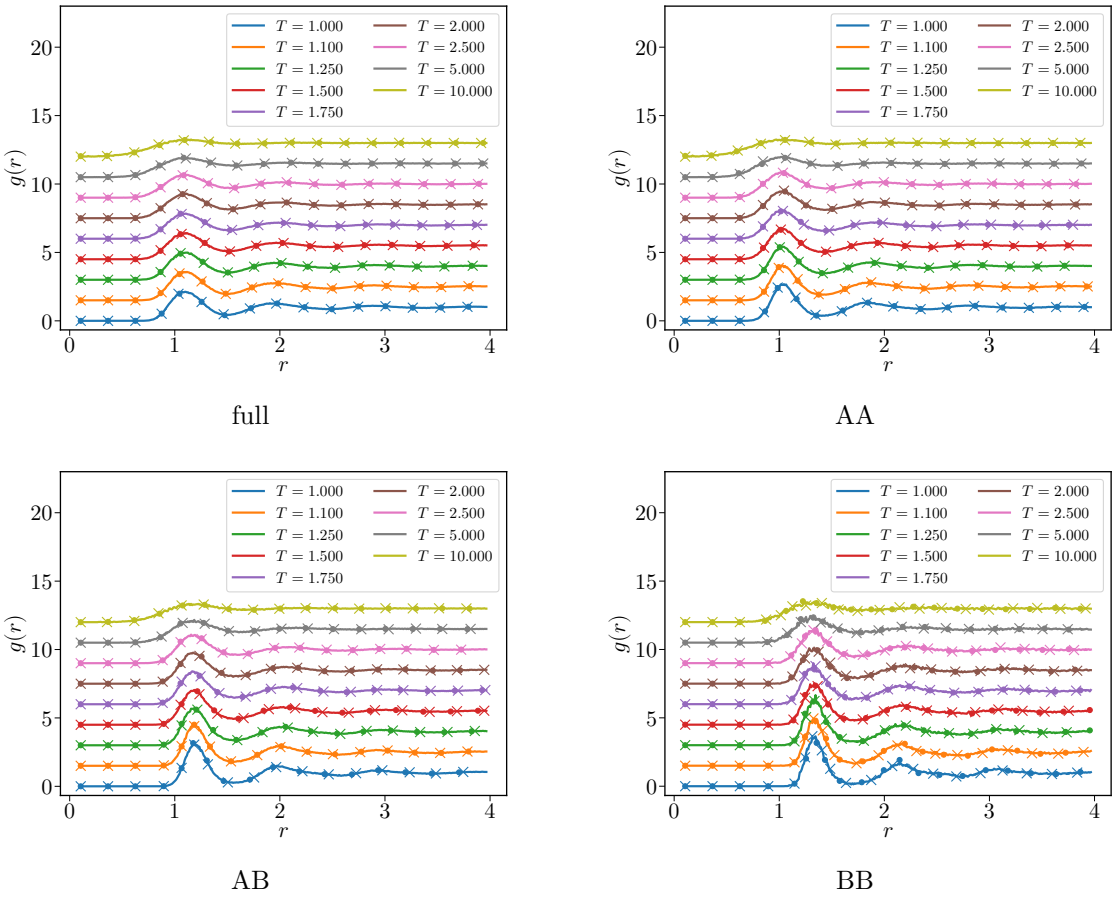


Figure 3.40: Radial distribution functions of the binary Voronoi fluid (solid lines) compared with simulations using NequIP with $l_{\text{max}} = 1$ (crosses) and $l_{\text{max}} = 2$ (dots) at density $\rho = 1.0$, concentration $\chi_{\text{A}} = 0.8$, and various temperatures T.

Self-Intermediate Scattering Functions

The dynamics reveal a more nuanced picture, as indicated by the self-intermediate scattering functions (SISFs). Results are shown in figs. 3.41 to 3.43 for $\chi_A = 0.2, 0.5, 0.8$, respectively. It should be noted that deviations in the SISFs can also arise from insufficient relaxation. Consequently, minor differences do not necessarily indicate deficiencies of the NNP. Unfortunately, due to the computational cost associated with NequIP, longer relaxation was not possible. For $\chi_A = 0.2$ and $\chi_A = 0.5$, both total and partial SISFs agree with the reference system, even in the regime of slow relaxation, with small deviations showing for the lowest temperature. However, at $\chi_A = 0.8$ and $T = 1.0$, significant discrepancies emerge, as neither the dynamics of the dominant A particles nor those of the B particles are reproduced. This behavior persists across multiple ensembles, ruling out statistical fluctuations as an explanation. Interestingly, the corresponding static properties show no significant anomaly, highlighting the sensitivity of slow dynamics to subtle differences in the PES.

Overall, mixtures with higher concentrations of B particles are captured more faithfully, consistent with the superior force reproduction for B particles. Deviations at $\chi_A = 0.8$ suggest that errors in the mixed interactions drive the observed differences, rather than pure A or B interactions. Moreover, systematic deviations between the two NequIP models are observed. The $l_{max} = 2$ network tends to overestimate relaxation times, while the $l_{max} = 1$ network underestimates them. Although both models reproduce the pure systems well (not shown), their accuracy in mixed systems with high A concentration appears limited, underlining the particular challenges of modeling slow dynamics with NNPs. A more rigorous test would require simulations at even lower temperatures, approaching dynamical arrest. However, due to the computational cost of NequIP, this was not feasible in the present work. Alternative architectures with improved parallelization may provide a path forward, enabling simulations of systems with significantly longer relaxation times.

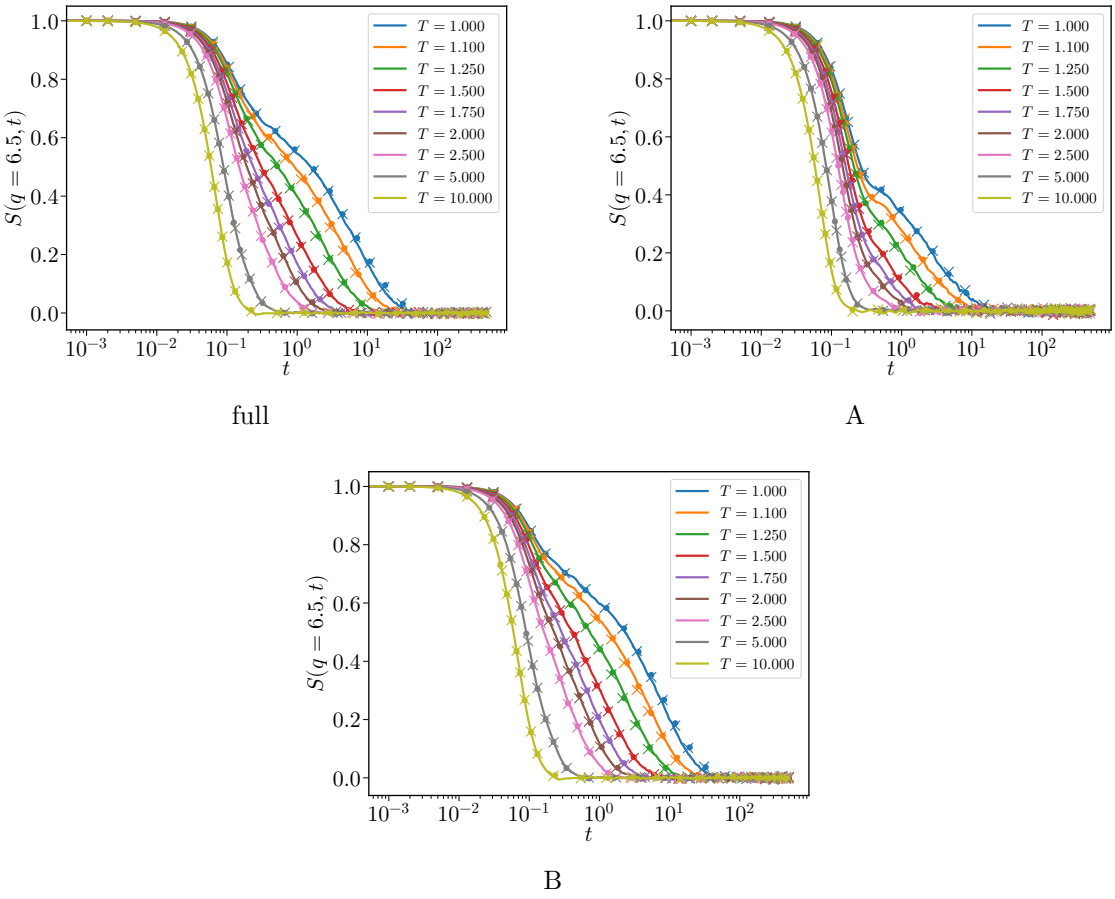


Figure 3.41: Self-intermediate scattering functions of the binary Voronoi fluid (solid lines) compared with simulations using NequIP with $l_{\max} = 1$ (crosses) and $l_{\max} = 2$ (dots) at density $\rho = 1.0$, concentration of A particles $\chi_A = 0.2$, and various temperatures T.

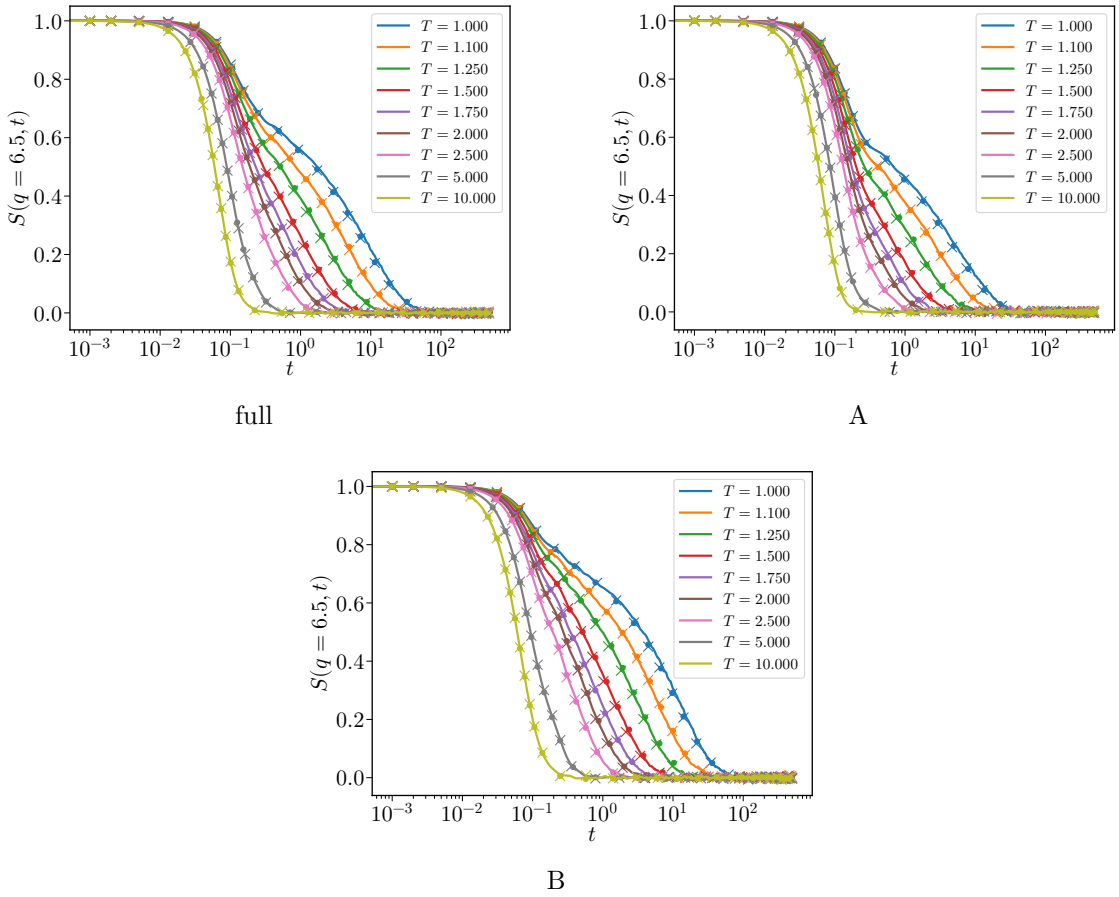


Figure 3.42: Self-intermediate scattering functions of the binary Voronoi fluid (solid lines) compared with simulations using NequIP with $l_{\text{max}} = 1$ (crosses) and $l_{\text{max}} = 2$ (dots) at density $\rho = 1.0$, concentration of A particles $\chi_{\text{A}} = 0.5$, and various temperatures T.

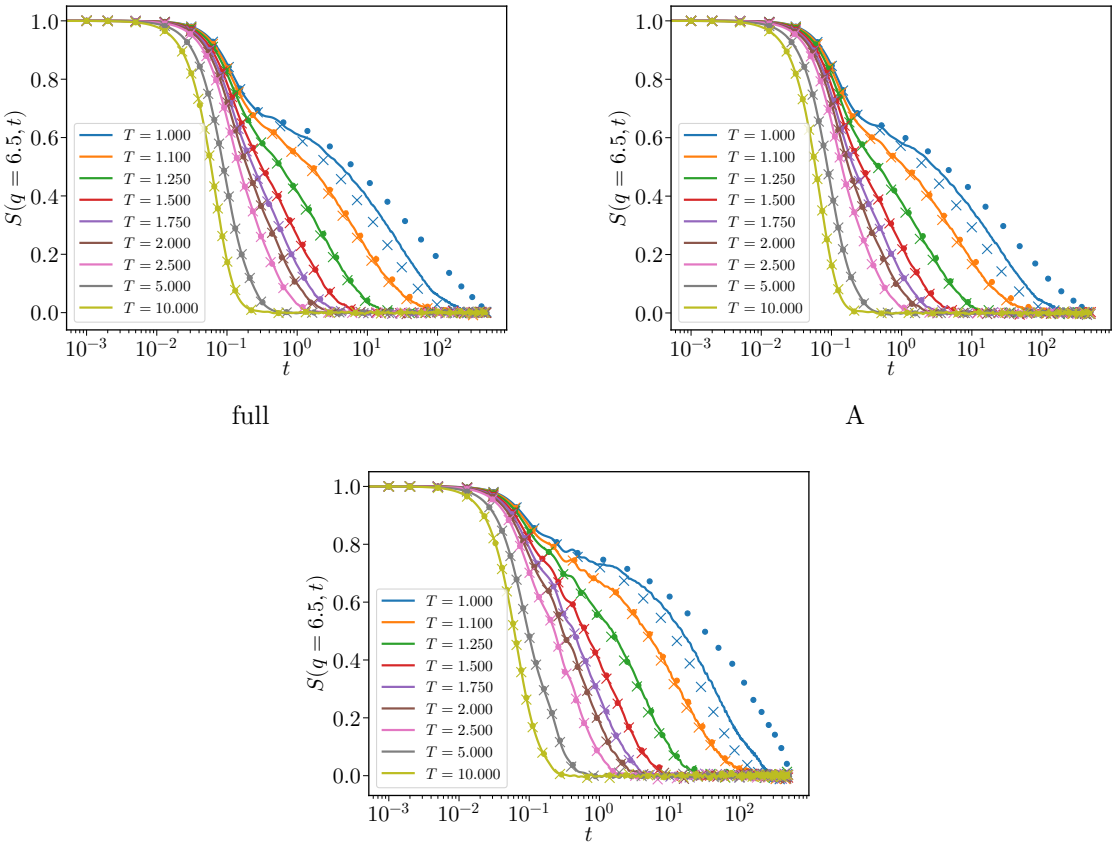


Figure 3.43: Self-intermediate scattering functions of the binary Voronoi fluid (solid lines) compared with simulations using NequIP with $l_{\max} = 1$ (crosses) and $l_{\max} = 2$ (dots) at density $\rho = 1.0$, concentration of A particles $\chi_A = 0.8$, and various temperatures T.

Summary

The investigation of the binary Voronoi potential highlights both the strengths and the limitations of NNPs in reproducing the dynamical behavior of glass-forming systems. Overall, both static and dynamic observables show very good agreement between the network potentials and the underlying Voronoi model across multiple state points. Nevertheless, deviations are identified in the regime closest to the glass transition, especially for systems with larger concentrations of small A particles. Here, the NNP failed to reproduce the dynamics observed in the reference simulations, despite convincing agreement in the static structure.

However, deviations cannot be attributed unambiguously to shortcomings of the potentials, as relaxation problems may also contribute. Due to the exponentially increasing computational effort required to equilibrate such systems, lower temperatures were not probed. Despite this, the observed disagreement appears to be inherent to the network potentials, since both NequIP models with $l_{\text{max}} = 1$ and $l_{\text{max}} = 2$ show comparable deviations, though in opposite directions. At the same time, both networks reproduce the dynamics of the other state points very well and were explicitly trained on configurations with even lower temperatures.

One possible explanation lies in the asymmetry of A and B particle contributions to the interactions. Because B particles contribute, on average, more to the potential energy, owing to their larger radii, and thus larger Voronoi cell volumes and moments of inertia, their influence persists even in local environments dominated by A particles. For local environments containing only B particles, this asymmetry disappears, and the network effectively learns two distinct internal energy redistributions for what should, in principle, reduce to the same monodisperse limits. Although these limits agree in terms of forces and total energies, the decomposition into atomic energy contributions differs between species. This inconsistency may, in turn, affect the behavior of mixed systems where both species are present.

While the binary Voronoi potential could be learned with high accuracy, the sensitivity of glassy dynamics to small deviations in the PES is also demonstrated. Subtle errors may amplify into significant differences in the long-time dynamics, underlining the importance of high-quality potentials. Although it is likely possible to train improved models that mitigate these issues to some degree, ultimately, the intrinsic fragility of glassy systems may set a fundamental limit to the reproducibility of their exact dynamics by approximate neural network potentials.

3.5 Boron

Having established the capabilities and limitations of NNPs in reproducing complex interaction models such as the Voronoi potential, as a short excursion, a NNP for a system based on DFT is trained. As an example, elemental boron is considered. Boron presents a particularly challenging case for NNPs due to its complex bonding environments. Capturing these subtleties requires a highly descriptive representation of the local atomic environment and robust generalization of the neural network architecture.

3.5.1 Training and Network Architectures

Initial investigations using AIMD trajectories and subsequent training of HDNNP models with the n2p2 package employing adaptive group lasso (AGL) regularization demonstrated the feasibility of constructing NNPs for boron [61]. These models, however, relied on energy-only training, as the AGL algorithm presented does not incorporate force information. While yielding reasonable energy predictions, this restriction left force errors systematically large.

Building on the experience with the Voronoi potential, more sophisticated architectures are used for training, namely DeePMD [15] and NequIP [17], incorporatating both energy and force information in the loss function and using a relative force weight $\mathcal{W}_\mathrm{F} = 10.0$. Training is performed on a large AIMD-generated dataset covering solid and liquid boron states at various temperatures [60]. Two HDNNP from the original work are taken for comparison, one using a full set of hand-crafted symmetry functions, and one with a reduced set selected via AGL [61].

The error statistics are summarized in tab. 3.9. The n2p2 models show low total energy errors, reflecting their training objective, but substantial force errors. DeePMD achieves a balanced performance, reducing force errors significantly while maintaining a competitive energy error. NequIP, particularly when including tensorial features up to rotational order $l_\mathrm{max} = 2$, further improves accuracy of force predictions while retaining small energy errors. The larger total energy error for $l_\mathrm{max} = 1$ is probably an outlier, based on the specific training run for the NNP. It is worth noting that the magnitude of the total energy per atom for the underlying boron configurations is typically in the order of $1 - 10\,\mathrm{eV}$.

Network	$\mathcal{W}_\mathrm{F}$	$E_\mathrm{RMSE,pa}$[meV]	F_RMSE[meV/Å]
n2p2$^\mathrm{ref}$ AGL	0	8.71	0.816
n2p2$^\mathrm{ref}$ full	0	7.78	0.734
DeePMD	10	9.61	0.457
NequIP $l_\mathrm{max} = 1$	10	21.10	0.357
NequIP $l_\mathrm{max} = 2$	10	9.31	0.219

Table 3.9: Different neural network potentials for boron. The reference n2p2 models are taken from [61]. The AGL network employs 6 radial and 10 angular symmetry functions, whereas the full model uses 12 radial and 48 angular symmetry functions.

3.5.2 Simulation Results

Simulations with the trained potentials are conducted for liquid boron across a representative range of temperatures at densities with approximately vanishing pressure. Due to some inconsistencies in data for the AIMD trajectories, dynamical observables could not be evaluated reliably, and the analysis is restricted to structural observables. The radial distribution function (RDF), $g(r)$, is chosen as the primary benchmark.

Fig. 3.44 shows RDFs obtained from the different potentials at select temperatures. The densities were determined by equilibrating the AIMD systems at vanishing pressure and subsequently fixing the density for NVT simulations. The qualitative agreement of peak positions across temperatures is convincing, with all models capturing the liquid structure. A more detailed comparison for the different temperatures is provided in fig. 3.45, which also highlights the first maximum and minimum of the RDF. These features are particularly sensitive to the short-range ordering characteristic of liquid boron.

The comparisons reveals clear trends. While the n2p2 models reproduce the overall RDFs reasonably well, deviations become apparent in the structure of the first coordination shell. DeePMD improves substantially upon this, reproducing the peak positions and amplitudes with better accuracy. NequIP delivers the best performance, with $l_\mathrm{max} = 1$ already sufficient to closely match the AIMD reference across all temperatures. Higher-order tensorial features, i.e., $l_\mathrm{max} = 2$, are not shown here. Nevertheless, based on prior observations, an even better match can be expected at the cost of higher computational demand.

To quantify these deviations, relative root mean squared errors between RDFs obtained from AIMD and those from simulations with the NNPs are shown in tab. 3.10. The superiority of NequIP is evident, outperforming both DeePMD and n2p2 across the full temperature range. Furthermore, the relative error decreases systematically with increasing temperature, reflecting the diminishing role of detailed potential features as the liquid approaches high temperatures.

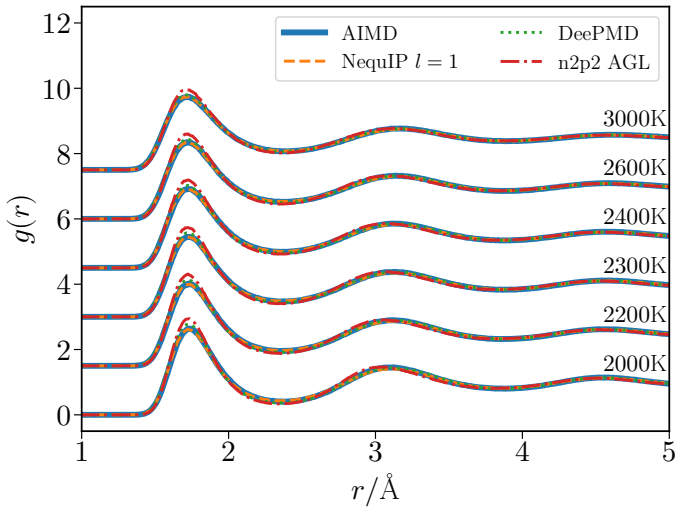


Figure 3.44: Comparison of radial distribution functions (RDF) for liquid boron obtained from AIMD simulations (solid) and from neural network potentials, NequIP with $l_{\text{max}} = 1$ (dashed), n2p2 with AGL (dash-dotted), and DeePMD (dotted), shown for different temperatures T.

T [K]	NequIP [%]	DeePMD [%]	n2p2$^{\text{ref}}$ full [%]	n2p2$^{\text{ref}}$ AGL [%]
2000	1.926	3.689	5.196	6.322
2200	1.333	3.037	4.456	5.841
2300	1.217	2.706	4.139	5.610
2400	1.117	2.420	3.954	5.368
2600	1.010	2.114	3.643	5.044
3000	0.779	1.549	3.502	4.368

Table 3.10: Root mean squared error of the radial distribution function $g(r)$ expressed as a percentage, for different neural network potentials of boron compared to reference AIMD data.

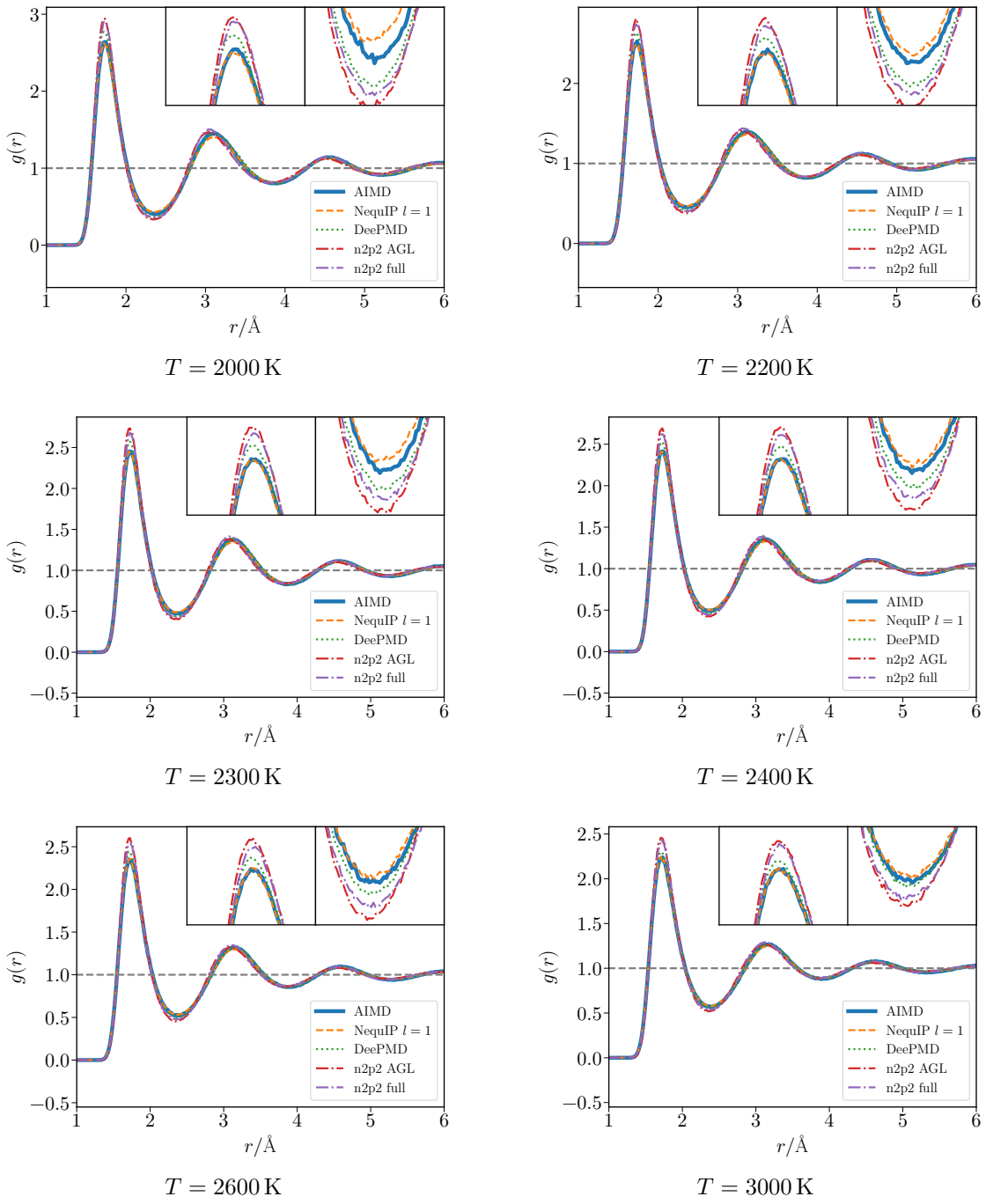


Figure 3.45: RDF comparison for liquid boron obtained from AIMD simulations (solid) and neural network potentials, NequIP with $l_{\text{max}} = 1$ (dashed), n2p2 with adaptive group lasso (dash-dotted), and DeePMD (dotted), shown for different temperatures T. Insets highlight the first maximum and first minimum.

Summary

These results demonstrate several important points. First, force-inclusive training is essential for constructing reliable potentials of real-world materials, as energy-only approaches, such as the AGL-trained HDNNP model, yield inferior accuracy in forces. Second, graph-based architectures such as NequIP clearly outperform symmetry-functions or embedding-based approaches, both in terms of error metrics and in reproducing physical observables. Third, for liquid boron, low order tensorial features, i.e., $l_{\mathrm{max}} = 1$, suffice to capture the essential structure.

Overall, this confirms that insights gained from model systems such as the Voronoi potential carry over to realistic materials. While the complexity of the local environment of DFT data requires powerful architectures, the combination of large datasets and modern neural network architectures makes accurate simulations possible.

3.6 Summary and Outlook

In this chapter, neural network potentials have been systematically investigated for two model systems of increasing complexity. Beginning with the Lennard-Jones potential and the glass-forming Kob–Andersen mixture, it was demonstrated that even comparatively simple NNPs can reproduce both static and dynamic properties with high accuracy, provided that the relevant regions of phase-space are sampled for training. Particular emphasis was placed on assessing potential quality beyond standard error metrics.

Building on this, the Voronoi potential was introduced as a more challenging test case. Its geometry-based interactions provided an excellent benchmark for different NNP architectures. Among the tested approaches, the graph-based model, NequIP, showed performance clearly superior to HDNNP and DeePMD, enabling stable and acurate simulations even for slow dynamics. The extension to the binary Voronoi system revealed some limitations of NNPs in capturing slow dynamics, as subtle asymmetries in species contributions and energy redistribution possibly translate into deviations near dynamical arrest. These limitations may not be inherent to NNPs themselves but show that even careful training can lead to an insufficient representation of the PES.

Finally, NNPs were applied to a reference system of liquid boron obtained from simulaton with density-functional theory, where NequIP again outperformed other approaches in reproducing static structure. This shows the capabilities of cutting-edge NNPs to bridge the gap between computationally expensive methods and large-scale molecular simulations. Taken together, the results of this chapter establish NNPs in the context of slow dynamics, providing a basis for their application to glass-forming systems.

In the future, for systems approaching dynamical arrest, alternative network architectures and training strategies could be explored to enhance performance near the glass transition, with particular emphasis on including configurations from both sides of the arrest in the training set. The general applicability of NNPs to the binary Voronoi system opens the door for computationally more expensive interactions. Whereas the complexity of the base model can be increased by introducing additional interaction terms, a NNP could capture the corresponding PES without major modifications or added computational cost. Although the binary Voronoi potential itself scales favorably for large systems, more intricate models may not be computationally feasible. In such cases, NNPs may provide a genuine performance advantage.

4. Neural Network Inverse Laplace Transform for the Calculation of Memory Kernels

The mode coupling theory of the glass transition (MCT) provides a microscopic framework to describe the dynamics of dense liquids and glass-forming systems [24]. It established a link between the static equilibrium structure and time-dependent relaxation processes. Despite its approximations, MCT remains one of the most successful theoretical approaches to glassy dynamics, offering both qualitative and quantitative insights into the physics [26].

The theoretical foundation of MCT relies on the Mori–Zwanzig (MZ) projection-operator formalism [89], which separates the dynamics into relevant and orthogonal subspaces. Applying this formalism to density fluctuations yields an integro-differential equation for the density correlation functions, the MZ equation, where a memory kernel encodes the structural relaxations. To render this equation solvable, the memory kernel is commonly approximated as a functional of the correlation functions themselves [24–26]. Once this memory kernel closure is specified, the MZ equation can be solved iteratively, producing time-dependent correlation functions and their associated memory kernels [90].

In experiments and simulations, correlation functions such as the intermediate and self-intermediate scattering functions are readily available, in contrast to the memory kernel, which is not directly measurable. While the MZ formalism does provide a relation that would allow calculating the memory kernel from a correlation function, numerical instabilities, behavior near the glass transition, and sensitivity to its short-time behavior render this calculation essentially impossible. A more approachable route transforms the MZ equation into Laplace space. Unfortunately, the critical step of performing the inverse Laplace transform (LT) to obtain the time-domain memory kernel remains numerically problematic, suffering from catastrophic cancellations, limited high-frequency resolution, and strong amplification of noise [27–29].

To overcome this challenge, the neural network inverse Laplace transform (NNLT) is introduced, trained on pairs of analytically known functions in the time and Laplace domain. This enables the neural network to generalize to physically relevant inputs without

relying on noisy or unknown transforms. Once trained, the network performs the inverse LT for Laplace-domain correlation functions arising in both theory and simulations, producing accurate time-domain representations.

One major complication arises from the unnormalized nature of memory kernels, compared to usually normalized correlation functions. To address this, a normalization scheme for the Laplace-domain memory kernel is developed, which ensures numerical stability of the NNLT and can also enforce physically meaningful scaling behavior. Together, the NNLT and normalization strategy establish a consistent and robust framework for the calculatation of memory kernels. The method is first validated using data from MCT, before being applied to simulations of Brownian particles.

With memory kernels available for Brownian dynamics simulations, the longstanding goal of deepening the understanding of the relationship between MCT and particle-based simulations can be pursued by constructing a functional relation between the memory kernel and the correlation function. In analogy to the mode coupling approximation, the memory kernel is expressed as a functional of the correlation functions. Using pairs of correlation functions and memory kernels, a feed-forward neural network is trained to approximate their relationship. Demonstrating feasibility allows to assess whether such a functional even exists for real-world particle systems and how it would differ from standard MCT.

This chapter is organized as follows. Sec. 4.1 introduces some theoretical foundations of MCT and the MZ equation. Sec. 4.2 presents the neural network inverse Laplace transform, which is trained on synthetic data and then validated on correlation functions from theory and simulations. In sec. 4.3, the NNLT is incorporated into a framework to compute time-domain memory kernels from the Laplace-domain Mori–Zwanzig equation, including a normalization procedure that tries to ensure consistency with physical scaling laws. The application to Brownian dynamics simulations is discussed in sec. 4.4, where memory kernels are obtained for noisy data and compared to MCT. Finally, sec. 4.5 introduces neural network functionals similar to the mode coupling approximation, which are trained on theoretical and simulation data, respectively. These functionals establish a new route to systematically compare theory and simulation and to probe the limitations of existing approximations.

4.1 Mode Coupling Theory of the Glass Transition

The starting point for deriving the microscopic equations of motion for a system of Brownian particles within mode coupling theory is the Fourier representation of the microscopic density for a system of N particles,

$$\rho(\boldsymbol{q}) = \frac{1}{\sqrt{N}} \sum_{j=1}^{N} \exp\bigl(\mathrm{i}\,\boldsymbol{q}\cdot\boldsymbol{r}_j\bigr)\,, \tag{4.1}$$

with wave vector $\boldsymbol{q}$ and particle positions $\boldsymbol{r}_j$. This quantity encodes fluctuations of particle positions and serves as the building block for the description of microscopic dynamics [24, 26].

The observable of interest is the density-density time correlation function,

$$S(\boldsymbol{q},t) = \bigl\langle \rho^*(\boldsymbol{q}) \exp(\Omega^\dagger t)\, \rho(\boldsymbol{q}) \bigr\rangle\,, \tag{4.2}$$

where Ω denotes the Smoluchowski operator governing the time evolution of the configurational probability distribution. Physically, $S(\boldsymbol{q},t)$ measures the persistence of density fluctuations over time and thus provides direct access to the relaxation dynamics of the liquid. In scattering experiments, it corresponds to the coherent dynamic structure factor, or intermediate scattering function (ISF), making it one of the most relevant experimentally accessible observables.

At $t = 0$, the ISF reduces to the static structure factor $S(\boldsymbol{q})$,

$$S(\boldsymbol{q}) = \langle \rho^*(\boldsymbol{q})\rho(\boldsymbol{q})\rangle, \tag{4.3}$$

which quantifies the equilibrium correlations of density fluctuations. For isotropic systems, $S(q)$ is related to the direct correlation function $c(q)$ through the Ornstein–Zernike equation [44, 91],

$$S(q) = \frac{1}{1-\rho c(q)}, \tag{4.4}$$

with number density $\rho = N/V$. This static structure factor serves as the sole input to MCT. The theory then provides a prediction for the full time evolution of $S(\boldsymbol{q},t)$, connecting microscopic structural information with the slowing down of relaxation upon approaching dynamical arrest.

4.1.1 Mori–Zwanzig Formalism

The Mori–Zwanzig projection operator formalism provides the theoretical foundation to derive equations of motion for time-dependent correlation functions such as $S(\boldsymbol{q},t)$ [89]. The microscopic dynamics are projected onto a sub-space spanned by relevant variables, in this case the density modes, thereby isolating these contributions from the fluctuations.

Adopting bra-ket notation, the equilibrium scalar product of two observables A and B is defined as

$$\langle A \,|\, B\rangle = \langle AB\rangle. \tag{4.5}$$

With this notation, the projection operator $\mathcal{P}$ and its orthogonal $\mathcal{Q}$ are introduced as

$$\mathcal{P} = \frac{\sum_{\boldsymbol{q}} |\rho(\boldsymbol{q})\rangle\langle\rho^*(\boldsymbol{q})|}{S(\boldsymbol{q})}, \tag{4.6}$$

$$\mathcal{Q} = 1 - \mathcal{P}, \tag{4.7}$$

satisfying the properties $\mathcal{P}^2 = \mathcal{P}$, $\mathcal{P}\mathcal{Q} = 0$, and $\mathcal{Q}|\rho(\boldsymbol{q})\rangle = 0$. Utilizing these projectors, the derivative of the time-evolution operator becomes

$$\partial_t \exp(\Omega^\dagger t) = \Omega^\dagger(\mathcal{P} + \mathcal{Q}) \exp(\Omega^\dagger t). \tag{4.8}$$

The next step is applying the Dyson decomposition [89],

$$\exp(\Omega^\dagger(\mathcal{P}+\mathcal{Q})t) = \exp(\Omega^\dagger \mathcal{Q} t) + \int_0^t \mathrm{d}t'\, \exp(\Omega^\dagger \mathcal{Q}(t-t'))\Omega^\dagger \mathcal{P} \exp(\Omega^\dagger t'), \tag{4.9}$$

and projecting onto density fluctuations by multiplying $\langle\rho^*(\boldsymbol{q})|$ from the left and $|\rho(\boldsymbol{q})\rangle$ from the right. This yields the Mori–Zwanzig equation of motion for the ISF,

$$\partial_t S(\boldsymbol{q},t) = -\frac{\omega(\boldsymbol{q})}{S(\boldsymbol{q})}\, S(\boldsymbol{q},t) + \frac{1}{S(\boldsymbol{q})} \int_0^t \mathrm{d}t'\, K(\boldsymbol{q}, t-t')\, S(\boldsymbol{q},t'), \tag{4.10}$$

with the frequency

$$\omega(\boldsymbol{q}) = -\left\langle \rho^*(\boldsymbol{q})\, \Omega^\dagger \rho(\boldsymbol{q})\right\rangle, \tag{4.11}$$

and diffusion kernel

$$K(\boldsymbol{q},t) = \left\langle \rho^*(\boldsymbol{q}) \Omega^\dagger \mathcal{Q} \exp(\Omega^\dagger t)\, \mathcal{Q}\Omega^\dagger \rho(\boldsymbol{q})\right\rangle. \tag{4.12}$$

4.1.2 Mori–Zwanzig Equation with Friction Kernel

It is often advantageous to reformulate the equation of motion in terms of a friction kernel rather than the diffusion kernel. Introducing another projection operator,

$$\mathcal{P}' = -\frac{\sum_{\boldsymbol{q}} |\rho(\boldsymbol{q})\rangle\langle\rho^*(\boldsymbol{q})\Omega^\dagger|}{\omega(\boldsymbol{q})}, \quad \mathcal{Q}' = 1 - \mathcal{P}', \tag{4.13}$$

and applying the Dyson decomposition from eq. (4.9) leads to an integro-differential equation for the diffusion kernel,

$$K(\boldsymbol{q},t) = M(\boldsymbol{q},t) - \frac{1}{\omega(\boldsymbol{q})}\int_0^t \mathrm{d}t'\, K(\boldsymbol{q},t-t')M(\boldsymbol{q},t'). \tag{4.14}$$

involving the friction kernel,

$$M(\boldsymbol{q},t) = \left\langle \rho^*(\boldsymbol{q})\Omega^\dagger \mathcal{Q} \exp\left(\Omega_\dagger \mathcal{Q}' t\right)\mathcal{Q}\Omega^\dagger \rho(\boldsymbol{q})\right\rangle. \tag{4.15}$$

Taking the Laplace transform of eq. (4.14) gives the relation

$$K(\boldsymbol{q},s) = \frac{M(\boldsymbol{q},s)}{1 + M(\boldsymbol{q},s)/\omega(\boldsymbol{q})}. \tag{4.16}$$

Substituting this into the Laplace transform of eq. (4.10), and transforming back to time domain, yields the Mori–Zwanzig equation of motion for the normalized ISF,

$$\tau(\boldsymbol{q})\,\partial_t \phi(\boldsymbol{q},t) + \phi(\boldsymbol{q},t) + \int_0^t \mathrm{d}t'\, m(\boldsymbol{q},t-t')\,\partial_{t'}\phi(\boldsymbol{q},t') = 0, \tag{4.17}$$

with $\phi(\boldsymbol{q},t) = S(\boldsymbol{q},t)/S(\boldsymbol{q})$, short-time relaxation time $\tau(\boldsymbol{q}) = S(\boldsymbol{q})/\omega(\boldsymbol{q})$, and memory kernel $m(\boldsymbol{q},t) = S(\boldsymbol{q})/\omega(\boldsymbol{q})^2\, M(\boldsymbol{q},t)$. This MZ equation constitutes the core of MCT needed for this work.

4.1.3 Mode Coupling Approximation

Deriving microscopic equations of motion for time-dependent correlation functions relies on finding an expression for the memory kernel arising within the Mori–Zwanzig formalism. In principle, the memory kernel contains contributions from correlations of arbitrarily high order, rendering the resulting equations of motion formally exact but practically unsolvable.

The defining step of MCT is the so-called mode coupling approximation, which provides a self-consistent closure of the memory kernel in terms of a functional of the density

correlation function,

$$m(\boldsymbol{q},t) = \mathcal{F}[\phi(\boldsymbol{q},t)]\,. \tag{4.18}$$

This assumes that the dominant contribution to the fluctuating forces comes from the coupling of a density mode at wave vector q to pairs of density fluctuations at other wave vectors, while contributions from higher-order correlations are neglected [26].

Usually, this approximation is implemented by projecting the fluctuating forces that define the memory kernel onto products of density modes. The resulting four-point correlation functions are then factorized into products of two-point correlators. As a consequence, the memory kernel becomes a quadratic functional of the correlation function [25],

$$m(\boldsymbol{q},t) \propto \int \mathrm{d}^2p\, \mathcal{V}^2(\boldsymbol{q},\boldsymbol{q}-\boldsymbol{p},\boldsymbol{p})\, S(\boldsymbol{q}-\boldsymbol{p},t)S(\boldsymbol{p},t), \tag{4.19}$$

with a vertex function $\mathcal{V}(\boldsymbol{q},\boldsymbol{k},\boldsymbol{p})$ that depends only on static correlations, such as the static structure factor or, equivalently, the direct correlation function [24].

This approximation transforms the MZ equation into a closed, self-consistent integro-differential equation for the ISF. The feedback loop between the correlation functions and their own memory kernel is responsible for the non-linear slowing down of relaxation and the emergence of glassy dynamics as described by MCT. The resulting theory has been shown to capture many qualitative, and in certain cases quantitative, features of glass-forming liquids [26].

Providing such a closure for the memory kernel enables numerical solutions of the MZ equation and, in schematic models, even analytic treatments. Practical implementations are often based on iterative schemes [90]. For instance, the open-source Python package mctpy [92] provides numerical routines for solving the MZ equation for a provided memory kernel functional.

4.1.4 Tagged-Particle Dynamics

Besides collective density fluctuations, the dynamics of a single particle embedded in the bulk system may also be of interest. The corresponding observable is the self-intermediate scattering function (SISF),

$$S^{\mathrm{s}}(\boldsymbol{q},t) = \left\langle \rho_{\mathrm{s}}^*(\boldsymbol{q}) \exp(\Omega^\dagger t)\, \rho_{\mathrm{s}}(\boldsymbol{q}) \right\rangle, \tag{4.20}$$

where $\rho_{\mathrm{s}}(\boldsymbol{q})$ denotes the density of the tracer. Using a similar projection-operator derivation for mixtures [93, 94], the MZ equation for $S^{\mathrm{s}}(\boldsymbol{q},t)$ is obtained, which after introducing the

memory kernel $m^{\mathrm{s}}(\boldsymbol{q},t)$ takes the form

$$\tau_{\mathrm{s}}(\boldsymbol{q})\,\partial_t S^{\mathrm{s}}(\boldsymbol{q},t) + S^{\mathrm{s}}(\boldsymbol{q},t) + \int_0^t \mathrm{d}t'\, m^{\mathrm{s}}(\boldsymbol{q},t-t')\,\partial_{t'} S^{\mathrm{s}}(\boldsymbol{q},t') = 0, \tag{4.21}$$

with $\tau_{\mathrm{s}}(\boldsymbol{q}) = 1/\omega^{\mathrm{s}}(\boldsymbol{q})$.

The mode coupling approximation yields a closure for the tracer memory kernel,

$$m^{\mathrm{s}}(\boldsymbol{q},t) \propto \int \mathrm{d}^2 p\, \mathcal{V}^{\mathrm{s}}(\boldsymbol{q},\boldsymbol{q}-\boldsymbol{p})^2\, S(\boldsymbol{q}-\boldsymbol{p},t)\, S^{\mathrm{s}}(\boldsymbol{p},t), \tag{4.22}$$

with vertex $\mathcal{V}^{\mathrm{s}}(\boldsymbol{q},\boldsymbol{k})$ including contributions from the direct correlation function between the tracer and the bulk system. The tracer dynamics are fully determined by the bulk structure factor $S(\boldsymbol{q})$ and the tracer-bulk correlations $S^{\mathrm{s}}(\boldsymbol{q})$, providing a microscopic description of single-particle motion near the glass transition.

4.1.5 Equilibrium Smoluchowski Dynamics

For a system of Brownian particles, overdamped dynamics are governed by the Smoluchowski operator

$$\Omega = \sum_i D_{\mathrm{t}} \nabla_i \left(\nabla_i - \beta \boldsymbol{F}_i\right), \tag{4.23}$$

which describes the stochastic evolution of the configurational probability distribution in terms of the translational diffusion coefficient D_{t} and the interparticle forces $\boldsymbol{F}_i$. Within the Mori–Zwanzig formalism, this determines the frequency

$$\omega(\boldsymbol{q}) = -\left\langle \rho^*(\boldsymbol{q})\, \Omega^\dagger \rho(\boldsymbol{q}) \right\rangle = D_{\mathrm{t}} q^2, \tag{4.24}$$

where $\rho(\boldsymbol{q})$ denotes the microscopic density. From this relation, the short-time relaxation time is obtained as

$$\tau(\boldsymbol{q}) = \frac{S(\boldsymbol{q})}{\omega(\boldsymbol{q})} = \frac{S(\boldsymbol{q})}{D_{\mathrm{t}} q^2}. \tag{4.25}$$

The mode coupling approximation introduces the vertex function [24],

$$\mathcal{V}(\boldsymbol{q},\boldsymbol{k},\boldsymbol{p}) = D_{\mathrm{t}} \left[(\boldsymbol{q}\cdot\boldsymbol{k})\, c(\boldsymbol{k}) + (\boldsymbol{q}\cdot\boldsymbol{p})\, c(\boldsymbol{p})\right], \tag{4.26}$$

where $c(\boldsymbol{q})$ is the direct correlation function. With this, the memory kernel can be expressed as a functional of density correlations, leading to a closed, non-linear integro-differential equation for the ISF.

The same reasoning can be extended to the dynamics of a single tracer particle embedded in the bulk [93, 94]. The short-time relaxation time then is

$$\tau_{\mathrm{s}}(\boldsymbol{q}) = \frac{1}{\omega^{\mathrm{s}}(\boldsymbol{q})} = \frac{1}{D_{\mathrm{t}}^{\mathrm{s}} q^2}, \tag{4.27}$$

where $D_{\mathrm{t}}^{\mathrm{s}}$ is the tracer diffusion coefficient. The corresponding vertex function is given by

$$\mathcal{V}^{\mathrm{s}}(\boldsymbol{q}, \boldsymbol{k}) = D_{\mathrm{t}}^{\mathrm{s}}(\boldsymbol{q} \cdot \boldsymbol{k})\, c^{\mathrm{s}}(\boldsymbol{k}), \tag{4.28}$$

with $c^{\mathrm{s}}(\boldsymbol{q})$ denoting the direct correlation function between tracer and bulk particles. The tracer memory kernel thus depends explicitly on tracer-bulk correlations, while the collective bulk dynamics enter through the (unnormalized) intermediate scattering function $S(\boldsymbol{q}, t)$. This provides a microscopic description of single-particle motion in dense systems.

To numerically solve the Mori–Zwanzig equation for Brownian particles, the static structure factor $S(\boldsymbol{q})$ and appropriate initial conditions are required. For isotropic systems, the wave vector $\boldsymbol{q}$ reduces to the wave number q. In a two-dimensional hard-disk system, the static structure factor can be obtained from fundamental-measure theory (FMT), which provides an explicit expression for the direct correlation function of a one-component fluid of particles with unit diameter [95].

The direct correlation function takes the form

$$\begin{aligned} c(q) = \frac{\pi}{6(1-\phi)^3 q^2} \Bigg[& -\frac{5}{4}(1-\phi)^2 q^2 \, J_0\Big(\frac{q}{2}\Big)^2 \\ & + \left(4(\phi(\phi-20)+7) + \frac{5}{4}(1-\phi)^2 q^2\right) \, J_1\Big(\frac{q}{2}\Big)^2 \\ & + 2(\phi-13)(1-\phi) q \, J_0\Big(\frac{q}{2}\Big) \, J_1\Big(\frac{q}{2}\Big) \Bigg], \end{aligned} \tag{4.29}$$

where $J_i(x)$ are Bessel functions of the first kind and $\phi = (\pi/4)\rho$ is the packing fraction. From this expression, the static structure factor $S(q)$ follows via the Ornstein–Zernike equation, see eq. (4.4).

To obtain a numerical solution for the MZ equation, a simple exponential $S(q,t) = \exp\left(-t/\tau(q)\right)$ is used as the first few values of the correlation function. Together with the static structure factor, this then provides the necessary input to solve the Mori–Zwanzig equation iteratively. The same direct correlation function is assumed for the interaction between bulk and the tagged-particle, i.e., $c^{\mathrm{s}}(q) = c(q)$. All numerical solutions of the MZ equation within this thesis are obtained with the mctpy package [92].

4.1.6 Schematic F_{12} Model

To introduce the central ideas of MCT in a simplified setting, it is instructive to consider the schematic F_{12}-model [96]. In this model, the dynamics of the correlation function, $S(t)$, are governed by a memory kernel closure of the form

$$m[S] = v_1 S(t) + v_2 S^2(t), \tag{4.30}$$

with phenomenological coefficients v_1 and v_2. This functional form captures the essential feedback mechanism underlying mode-coupling theory.

With this closure specified, the Mori–Zwanzig equation can be solved iteratively to obtain both the correlation function and the corresponding memory kernel. Despite its simplicity, the F_{12}-model reproduces central features of glassy dynamics, including the characteristic two-step relaxation process with an intermediate plateau, slow structural relaxation, and the bifurcation associated with the glass transition.

The physically admissible parameter range for the coefficients v_1 and v_2 follows from the long-time limit of the correlation function, which defines the non-ergodicity parameter

$$f = \lim_{t\to\infty} S(t). \tag{4.31}$$

This parameter quantifies the degree of dynamical arrest [97, 98] and satisfies the MCT bifurcation relation

$$\frac{f}{1-f} = m[f] = v_1 f + v_2 f^2. \tag{4.32}$$

A non-zero solution $f > 0$ signals a glassy state. Solving the bifurcation equation yields the critical line separating ergodic liquid and glassy regimes,

$$v_2^{\mathrm{c}} = \frac{(1-v_1^{\mathrm{c}})^2}{4}, \qquad v_1^{\mathrm{c}} = 2\sqrt{v_2^{\mathrm{c}}} - v_2^{\mathrm{c}}, \tag{4.33}$$

with physically meaningful solutions for $v_1 \geq 0$ and $v_2 \geq 0$. Within this domain, the model reproduces characteristic features of glass-forming systems, including the emergence of critical power- law regimes near the transition and divergence of relaxation times. Because of this, the F_{12} model is suitable for exploring and validating new theoretical approaches within the framework of MCT.

4.2 Neural Network Inverse Laplace Transform of Correlation Functions

In Laplace space, the Mori–Zwanzig equation provides an algebraic connection between a correlation function and its associated memory kernel. This relation is commonly expressed in the standard form

$$\widehat{S}(q,s) = \frac{1}{s + \dfrac{1}{\tau(q) + \widehat{m}(q,s)}}. \tag{4.34}$$

Obtaining the memory kernel in the time domain requires an inverse Laplace transform. While this suggests a straightforward computational route, the inverse LT is well known to exhibit substantial numerical instabilities [27–29]. Direct inversion methods amplify statistical noise, suffer from catastrophic cancellations, and require high-frequency spectral data that is often unavailable in simulations or experiments. In the following, without loss of generality, the Laplace transform will be evaluated on the imaginary axis with $s = \mathrm{i}\omega$.

To address these issues, the neural network inverse Laplace transform (NNLT) is introduced. The aim of this method is to approximate the mapping from a function sampled in Laplace space, $\{\widehat{f}(\omega_1), \ldots, \widehat{f}(\omega_N)\}$, to its time domain counterpart, $\{f(t_1), \ldots, f(t_N)\}$, defined on fixed grids in both ω and t. The network is implemented as a simple feed-forward architecture, serving as a universal function approximator capable of learning this non-linear mapping directly from data.

For stability, the focus is placed on normalized correlation functions restricted to the interval $f(t) \in [0, 1]$. This normalization reduces sensitivity to high-frequency components, which otherwise dominate the short-time regime and are most prone to numerical error. In the case of memory kernels, which are not normalized, an explicit normalization or specification of the initial value is required in any case, so this restriction poses no limitation.

The success of the neural network depends critically on the construction of suitable training data. To this end, analytic function pairs representing both the time domain correlation function and its LT are generated. Superpositions of decaying exponentials with broadly distributed relaxation times are chosen, as they naturally capture the relaxation processes observed in glassy systems [98, 99]. Since both $f(t)$ and $\widehat{f}(\omega)$ can be evaluated exactly, the network is able to learn the true inverse LT evaluated on a grid, instead of a discretized transform.

Several features of the NNLT and its training process are motivated by physical and numerical considerations. neural networks can provide superior numerical stability compared to direct inversion methods, particularly in the presence of noise. Unlike conventional

techniques that require complete, noise-free spectra, the network can be trained to operate on truncated or incomplete input, a significant advantage given that high-frequency components are often unreliable. By restricting the output to normalized functions, the network enforces the initial condition $f(0) = 1$, eliminating the need to separately encode it. The architecture can be tailored to reproduce exponential-like decay as $f(t) \to 0$, for example through activation functions such as tanh or sigmoid. Furthermore, the conjugate symmetry of the LT of real-valued functions can be leveraged by restricting the input to the positive frequency-domain, thereby reducing the dimensionality of the input space. Finally, logarithmically spaced grids in both ω and t ensure efficient resolution across the time scales characteristic of structural relaxation.

Taken together, these design choices provide a robust framework for approximating the inverse LT of correlation functions. By combining analytic training data with the flexibility of neural networks, the NNLT overcomes the numerical problems of direct inversion schemes, enabling reconstruction of time-domain functions and, ultimately, memory kernels from their Laplace-space representations.

4.2.1 Network Architecture

The NNLT is implemented as a simple feed-forward neural network using PyTorch [53]. Its input consists of the positive frequency spectrum of Laplace-domain functions evaluated along the imaginary axis, $\hat{f}(\omega)$. Owing to conjugate symmetry, negative frequencies are discarded without loss of information. The output is the time-domain function $f(t)$. For correlation functions represented as superpositions of monotonically decaying exponentials, $f(t)$, the Laplace transform satisfies

$$\operatorname{Re} \hat{f}(\omega) > 0, \qquad \operatorname{Im} \hat{f}(\omega) < 0. \tag{4.35}$$

The frequency grid of the input and time grid of the output are chosen to be logarithmically spaced, aligning with the time-scales for typical correlation functions. This choice, while not strictly necessary, can substantially improve network performance. The smallest frequency is determined by the largest time point in the t-grid defined for the network, while the upper limit is set by the stability threshold of the Filon method for the numerical Fourier–Laplace transform, see sec. 2.3.2, typically $\omega \lesssim 10^2 - 10^4$.

To prepare the input vector of the network, $\hat{f}(\omega)$ can be represented either in polar form,

$$\left\{\log|\hat{f}(\omega_i)|,\, \angle\hat{f}(\omega_i)\right\}_i, \tag{4.36}$$

or in Cartesian form,

$$\left\{\log \mathrm{Re}(\hat{f}(\omega_i)),\, \log \mathrm{Im}(-\hat{f}(\omega_i))\right\}_i. \tag{4.37}$$

Both encodings show similar performance for decaying exponentials, with polar coordinates tending to be more robust under noisy conditions and for function values close to zero. For oscillatory correlation functions, where the sign of $\mathrm{Im}\,\hat{f}(\omega_i)$ may change, polar representation becomes essential. The input tuples are concatenated into a single vector, which is then passed to the neural network. For noisy data, unphysical function values are truncated to ensure that the input to the logarithms remains bounded and polar angles are restricted to the range occurring for this family of functions.

The network itself consists of N fully connected layers with l_i neurons each, with shallow networks proving sufficient. The size of the hidden layers is chosen to be of the same order as the number of input values and tanh activations between layers. The output layer has the same dimension as the number of points in the time grid, with suitable activation functions enforcing normalization. A sigmoid is employed when $f(t) \in [0,\, 1]$, while tanh is used when $f(t) \in [-1, 1]$. These constraints eliminate the need to learn arbitrary prefactors and ensure the correct short-time behavior even when high-frequency information is missing. Moreover, the scaling properties of these activations enable the network to reliably resolve values as small as $f(t) \approx 10^{-9} - 10^{-12}$.

The resulting output corresponds to the desired inverse LT,

$$f(t) = \mathcal{L}^{-1}\{\hat{f}(\omega)\}, \tag{4.38}$$

evaluated on a logarithmic time grid, designed to match the relaxation windows encountered in Brownian dynamics simulations and MCT.

Alternative architectures were explored in which the network directly predicts coefficients a_k of a fixed exponential basis,

$$f(t) \approx \sum_k a_k \exp(-\lambda_k t), \tag{4.39}$$

with predetermined exponents λ_k. While this approach can improve robustness against noise, it suffers some severe drawbacks. These functions fails to vanish at long times due to coefficients a_k not becoming exactly zero. Additionally, the restricted functional form fails to capture non-exponential scaling laws such as those found in the asymptotic regime of the F_{12} model. For this reason, direct prediction of the function values on a grid is preferred.

Training is performed on synthetic data with analytically known LT. Only functions that decay from $f(0) \approx 1$ to values near zero within the time window $[t_{\min}, t_{\max}]$ are included, ensuring that the inverse transform is well-defined and that the network is exposed to only fully decaying functions.

4.2.2 Training Data Generation

The performance of the NNLT depends on accurate and representative training data. Accordingly, the dataset is constructed from the same functional family as the target correlation functions. For Brownian particles, i.e., a system obeying Smoluchowski dynamics, correlation functions are completely monotone and can be written exactly as superpositions of decaying exponentials [98], following a distribution of relaxation times [99]:

$$f(t) = \sum_{k=1}^{N} a_k \exp(-t/\tau_k)\,. \tag{4.40}$$

For this class of functions, the LT is known analytically, see sec. 2.3.1,

$$\hat{f}(\omega) = \sum_{k=1}^{N} \frac{a_k}{\mathrm{i}\omega + 1/\tau_k}. \tag{4.41}$$

This choice enables training on exact analytical function pairs $(f(t),\, \hat{f}(\omega))$.

Each training function is generated by sampling the number of terms N from a uniform integer distribution, typically $N \in [1, 200]$. The amplitudes a_k are drawn from a uniform distribution subject to the constraint $\sum_k a_k = 1$, which guarantees $f(0) = 1$. The relaxation times τ_k are bounded such that $f(t)$ decays to near zero at $t_{\max}$, and are distributed on a logarithmic grid spanning $[\tau_{\min}, \tau_{\max}]$, ensuring that relaxation processes across the full time window are sampled.

Several sampling strategies for τ_k were tested, including uniform, unimodal-normal, and bimodal-normal distributions. A bimodal distribution of logarithmic relaxation times proved most effective, yielding the best performance when evaluating the trained network on actual correlation functions. It also closely captures the separation of time scales characteristic of glassy systems. The means of the two modes are chosen as $m_1 = 1$, corresponding to the Brownian time scale with $D_\mathrm{t} = 1$, and $m_2 = 10^4$, motivated empirically, while the widths σ_1 and σ_2 are sampled randomly in the order m_i. Representative examples of the generated training pairs are shown in fig. 4.1.

The final dataset consists of a large collection of function pairs evaluated on logarithmically spaced grids in both $t \in [t_{\min}, t_{\max}]$ and $\omega \in [\omega_{\min}, \omega_{\max}]$. Functions that decay either too rapidly or too slowly due to statistical tails of the distributions are discarded.

Constructing analogous datasets for active Brownian particles (ABPs) proved more challenging. A suitable functional basis capable of capturing the oscillatory and non-trivial relaxation dynamics of dense active systems could not be identified within the scope of this work. Initial attempts using superpositions of free-particle solutions successfully reproduced correlation functions in the dilute regime, but failed to capture complex oscillatory features of dense ABP systems.

To overcome these difficulties in the future, more general training sets could be constructed by incorporating additional functional forms, such as oscillatory terms, stretched exponentials, or power-law decays. For such functions, the LT would need to be computed numerically. Further improvements could be achieved by adding controlled levels of noise or drift. Optimizing the sampling of ω-values on the logarithmic grid, for example by concentrating points at low frequencies to resolve long-time decay or at high frequencies to capture short-time motion, could improve the accuracy of the trained network.

For Brownian particles, the bimodal exponential basis provides a reliable foundation for training the NNLT, while future extensions to richer function classes will be necessary to capture the full diversity of correlation functions encountered in active or more complex systems.

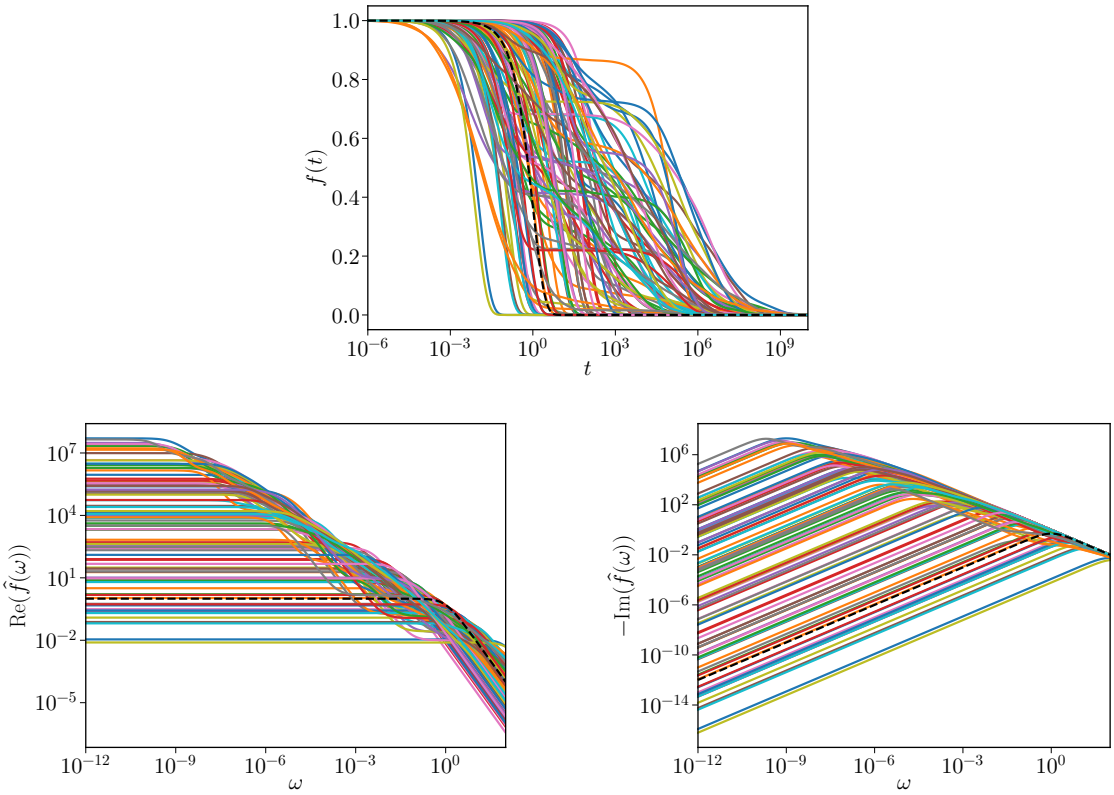


Figure 4.1: Examples of superpositions of exponential functions (top) and their Laplace transforms (bottom), evaluated on the imaginary axis with relaxation times drawn from bimodal log-normal distributions. These function pairs form the training data for the NNLT.

4.2.3 Training Procedure

The training procedure is tailored to the data to achieve good performance for the NNLT. Optimization is performed using mini-batch stochastic gradient descent with ADAMW [47], combined with an exponentially decaying learning rate to improve convergence. The network is trained on randomly sampled function pairs in batches of size 128, motivated by balancing gradient stability against th overall training time.

The loss function $\mathcal{L}$ is designed to capture both absolute and relative errors, consisting of two terms. A standard mean squared error (MSE) penalizes discrepancies between predicted and true values on a linear scale, while a logarithmic MSE compares the logarithms of predictions and targets. The latter is introduced to emphasize the asymptotic regime of correlation functions, where function values become small but remain significant. A small offset $\epsilon = 10^{-12}$ is added to arguments of the logarithms to ensure numerical stability. Explicitly, the loss is defined as

$$\mathcal{L}_{\mathrm{MSE}} = \frac{1}{N}\sum_i \left(y_{\mathrm{pred},i} - y_{\mathrm{true},i}\right)^2, \tag{4.42}$$

$$\mathcal{L}_{\mathrm{MSE,log}} = \frac{1}{N}\sum_i \left(\log(y_{\mathrm{pred},i} + \epsilon) - \log(y_{\mathrm{true},i} + \epsilon)\right)^2, \tag{4.43}$$

$$\mathcal{L} = \mathcal{W}_{\mathrm{MSE}}\,\mathcal{L}_{\mathrm{MSE}} + \mathcal{W}_{\mathrm{MSE,log}}\,\mathcal{L}_{\mathrm{MSE,log}}. \tag{4.44}$$

The weights $\mathcal{W}_{\mathrm{MSE}}$ and $\mathcal{W}_{\mathrm{MSE,log}}$ determine the relative importance of each contribution and are tuned to optimize performance.

The inclusion of the logarithmic loss term is crucial to reproduce the asymptotics of correlation functions. Without this contribution, the network systematically underestimates long-time decay, effectively truncating the tail of the correlation function. The logarithmic term, however, is only applicable to strictly positive functions and thus cannot be employed directly for sign-alternating correlation functions, such as those encountered in active systems.

Possible future improvements include adaptive weighting schemes for $\mathcal{W}_{\mathrm{MSE}}$ and $\mathcal{W}_{\mathrm{MSE,log}}$, dynamic adjustment of the batch size, or staggered training procedures in which the network is first exposed to simple exponential decays and subsequently to more complex functional forms. Such refinements could further enhance the generality and robustness of the NNLT across a broader class of correlation functions.

4.2.4 Network Training

With the dataset generation and training procedure specified, the NNLT can be trained. For the final training, approximately 50 000 function pairs are generated according to the procedure described in sec. 4.2.2, split into a training and a validation set. The input is represented on a logarithmic frequency grid of 500 points spanning $\omega \in [10^{-12}, 10^{2}]$. The corresponding time-domain output is evaluated on a logarithmic grid of 500 values in $t \in [10^{-6}, 10^{10}]$. As discussed in sec. 4.2.1, the input is encoded in polar form to enhance numerical stability.

Systematic tests demonstrated that the predictive accuracy of the network is largely robust against variations in grid resolution, activation function choice, and moderate changes in model depth and width. Even with halved input and output resolution, comparable accuracy can be obtained. Architectures with at least two hidden layers and widths on the order of the number of time-grid points proved sufficient. Increasing depth or width beyond this range does not improve accuracy and frequently led to overfitting. A residual-network (ResNet) variant was also tested but showed no advantage over simpler feed-forward designs.

Based on these observations, a standard architecture with two hidden layers of 1024 neurons each is selected. This configuration provided the best trade-off between accuracy, training stability, and generalization to unseen data. Alternative models, including smaller networks with 512 neurons per layer, as well as larger models with three layers of 2048 neurons or four layers of 1024 neurons, were tested but yielded no significant improvement. The training performance of the final networks is illustrated in fig. 4.2 with corresponding error metrics summarized in tab. 4.1.

In summary, the results show that relatively simple feed-forward architectures suffice for the NNLT, provided that appropriate data encoding, normalization, and training strategies are employed.

Loss	Standard	2×512	2×2048	4×1024	ResNet
$\mathcal{L}$	$6.120 \cdot 10^{-5}$	$8.111 \cdot 10^{-5}$	$2.955 \cdot 10^{-5}$	$3.283 \cdot 10^{-5}$	$8.714 \cdot 10^{-5}$
$\mathcal{L}^{\mathrm{val}}$	$6.435 \cdot 10^{-5}$	$9.140 \cdot 10^{-5}$	$4.654 \cdot 10^{-5}$	$5.376 \cdot 10^{-5}$	$1.021 \cdot 10^{-4}$
$\mathcal{L}_{\mathrm{MSE}}$	$6.485 \cdot 10^{-6}$	$1.041 \cdot 10^{-5}$	$2.895 \cdot 10^{-6}$	$4.108 \cdot 10^{-6}$	$1.034 \cdot 10^{-5}$
$\mathcal{L}^{\mathrm{val}}_{\mathrm{MSE}}$	$8.444 \cdot 10^{-6}$	$1.739 \cdot 10^{-5}$	$1.067 \cdot 10^{-5}$	$1.996 \cdot 10^{-5}$	$2.146 \cdot 10^{-5}$
$\mathcal{L}_{\mathrm{MSE,log}}$	$5.472 \cdot 10^{-5}$	$7.070 \cdot 10^{-5}$	$2.665 \cdot 10^{-5}$	$2.872 \cdot 10^{-5}$	$7.680 \cdot 10^{-5}$
$\mathcal{L}^{\mathrm{val}}_{\mathrm{MSE,log}}$	$5.590 \cdot 10^{-5}$	$7.402 \cdot 10^{-5}$	$3.587 \cdot 10^{-5}$	$3.380 \cdot 10^{-5}$	$8.067 \cdot 10^{-5}$

Table 4.1: Error metrics for the trained NNLTs, corresponding to the training results illustrated by the histograms in fig. 4.2. The table reports both linear and logarithmic mean squared errors.

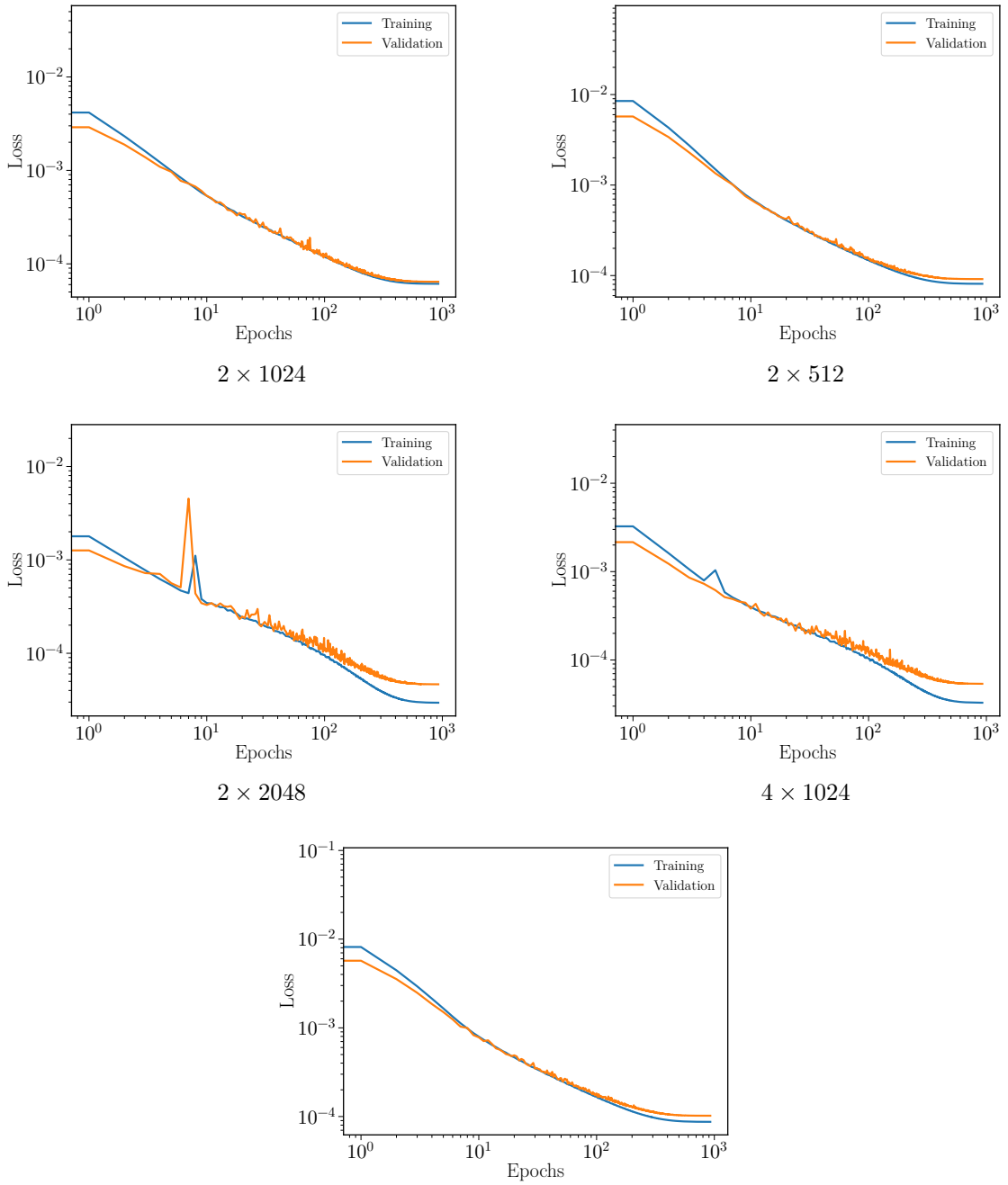


Figure 4.2: Histogram of total losses during training of NNLTs for several network architectures. An exponentially decaying learning rate is employed, and convergence is observed for all cases. Training errors are computed as running averages over one epoch.

4.2.5 Network Testing and Validation

To assess the performance of the NNLT beyond synthetic training data, the network is tested on physically relevant correlation functions obtained from both mode coupling theory and Brownian dynamics simulations. While training relied on superpositions of exponential decays with analytically known LT, the objective here is to evaluate the ability of the network to generalize and reliably perform the inverse LT of real-world correlations functions.

For theoretical benchmarks, correlation functions are generated numerically within mode coupling theory with the mctpy package [92]. In particular, the F_{12} schematic model and the two-dimensional Brownian hard-disk model based on fundamental-measure theory (FMT) are of interest. The functions are transformed numerically into Laplace domain using the Filon method, see sec. 2.3.2, which provides accurate Laplace-domain representations within its range of stability.

The evaluation is further extended to data generated by Brownian dynamics simulations, see sec. 2.1.2, which necessarily contain noise. The presence of noise poses a challenge for the NNLT, since statistical fluctuations are most pronounced at high frequencies and can severely affect short-time behavior. Nevertheless, the Filon method remains sufficiently accurate, provided that unstable high-frequency contributions are truncated. Applying the trained NNLT to these Laplace-transformed correlation functions enables direct comparison with the original time-domain data, quantifying the ability of the network to generalize to noisy input.

A particular focus of the validation lies in non-exponential asymptotic behavior. Such features are difficult to represent with finite sums of exponentials, making them a stringent test of the capacity of the network to capture slow relaxation processes. Systematic comparison of the predicted and true asymptotics provides valuable insight into the reliability of the method.

It should be emphasized that deviations between reconstructed and reference correlators may also originate from inaccuracies of the initial LT itself rather than from the neural network. However, by restricting the analysis to frequency ranges in which the Filon method is numerically stable, such artifacts can be minimized.

In the following, results of the NNLT are compared to original correlation functions from the F_{12} model, the Brownian disk model, and simulations of two-dimensional Brownian particles, providing a comprehensive assessment of network performance.

F_{12}-Model

The schematic F_{12} model provides a minimal, wave number-independent framework for studying glassy dynamics, see sec. 4.1.6. Despite its simplicity, the model reproduces the essential features of the glass transition, including the bifurcation and qualitative scaling laws of mode coupling theory. It is therefore ideally suited for validating the NNLT.

Two limiting cases are particularly interesting. The F_2 model ($v_1 = 0$) exhibits glassy two-step relaxation with a pronounced plateau, while the F_1 model ($v_2 = 0$) reduces to simple exponential decay. These complementary scenarios enable a stringent evaluation of network performance across distinct dynamical regimes.

For the F_2 model, see fig. 4.3, the correlation functions reconstructed by the NNLT closely match the original time domain input, even though only truncated frequency information was provided. The network successfully reproduces the long-time exponential decay, with discrepancies appearing only near machine precision. Small deviations are also observed for correlation functions with extremely long plateaus, where decay times extend to the limits of the time grid. These deviations are attributed to the underrepresentation of such slow-relaxing functions in the training set and not a systematic failure of the method.

For the simpler F_1 model (fig. 4.4), the NNLT can reproduce the correlation functions well. Minor deviations in the long-time regime can be observed but have negligible influence on the overall agreement. Together, these results demonstrate the ability of the network to perform the inverse LT for correlation functions across a wide range of relaxation characteristics. The accuracy of the Filon method for the forward transform ensures that discrepancies originate from the inversion itself, except possibly in the regime where values approach numerical precision limits.

Beyond convincing reproduction of the time domain correlation functions, the F_{12} model also enables a test of the ability of the network to capture asymptotic scaling near the glass transition. Near the critical point ϕ^c, the correlation function exhibits two universal power laws,

$$\phi(t) - \phi^c \sim t^{-a}, \quad \text{(critical decay)}, \tag{4.45}$$

$$\phi(t) - \phi^c \sim -t^b, \quad \text{(von Schweidler law)}, \tag{4.46}$$

where the exponents a and b are determined by a parameter λ [92].

Fig. 4.5 and fig. 4.6 illustrate the neural network predictions of the asymptotic scaling laws close to the glass transition. The network reproduces both regimes with good accuracy, including the correct exponents, confirming its capability to learn non-exponential behavior and non-trivial power laws not explictily present in the training data. The largest deviations

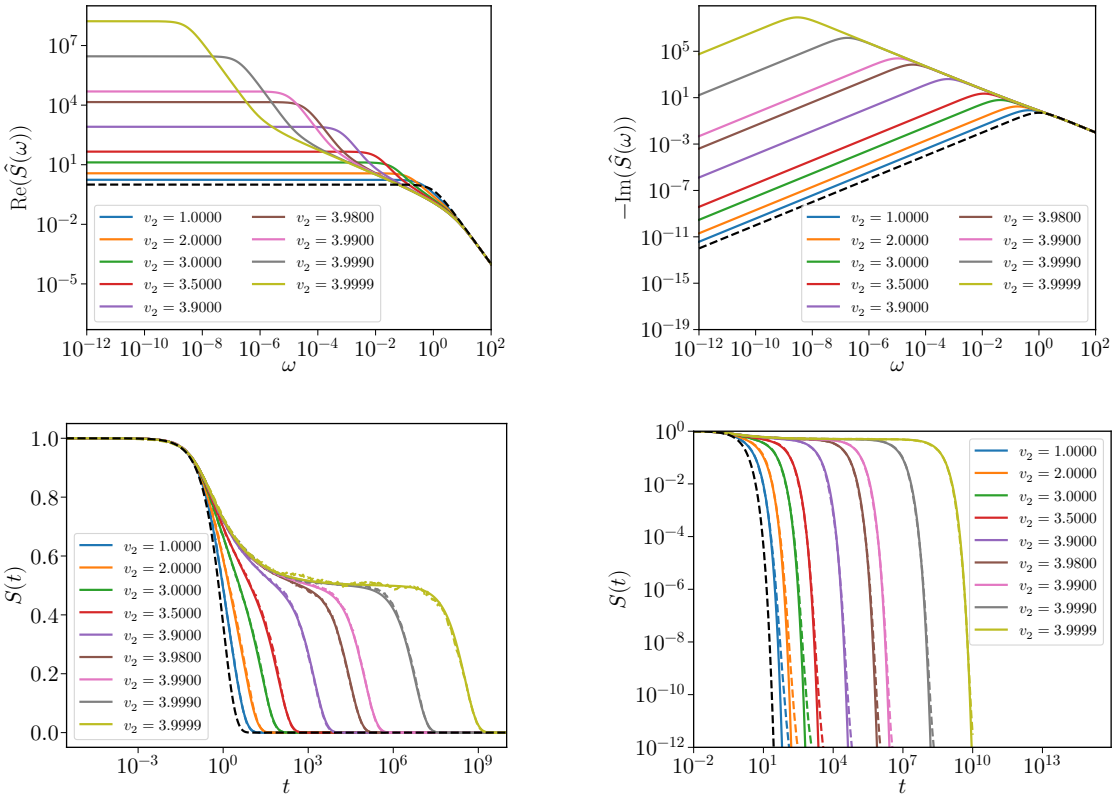


Figure 4.3: Results for the inverse Laplace transform of correlation functions for the F_2 model. The top panels display the Laplace-domain correlation functions obtained via the Filon method applied to numerical solutions of the Mori–Zwanzig equation. The bottom panels compare the corresponding time-domain correlation functions. Solid lines denote the numerical solutions, while dashed lines show the output of the NNLT.

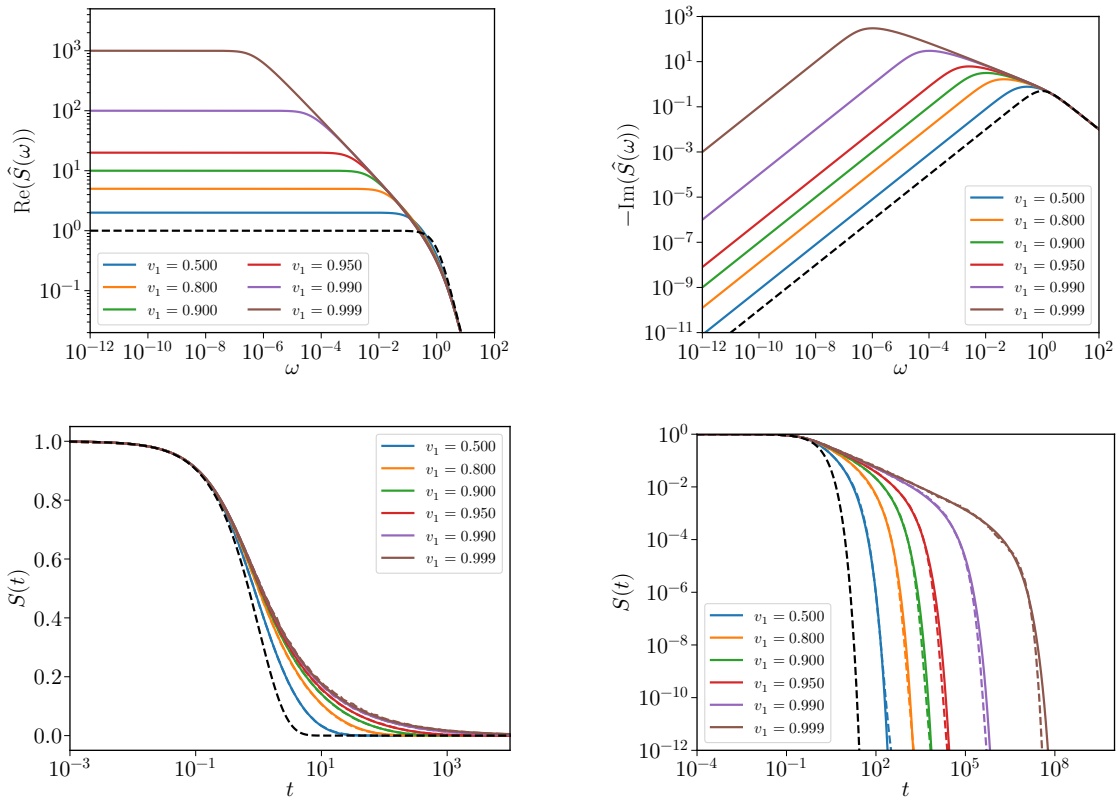


Figure 4.4: Results for the inverse Laplace transform of correlation functions for the F_1 model. The top panels display the Laplace-domain correlation functions obtained via the Filon method applied to numerical solutions of the Mori–Zwanzig equation. The bottom panels compare the corresponding time-domain correlation functions. Solid lines denote the numerical solutions, while dashed lines show the output of the NNLT.

occur for parameter $v_2 = 3.999$, where plateau times are longest and the correlation functions appear to be underrepresented in the training data. Nonetheless, even in this extreme case the reconstructed correlation functions retain the correct qualitative behavior.

In summary, the results for the F_{12}-model show that the NNLT not only faithfully reconstructs time domain correlation functions from frequency-space input, but also captures the universal asymptotic laws of glassy dynamics.

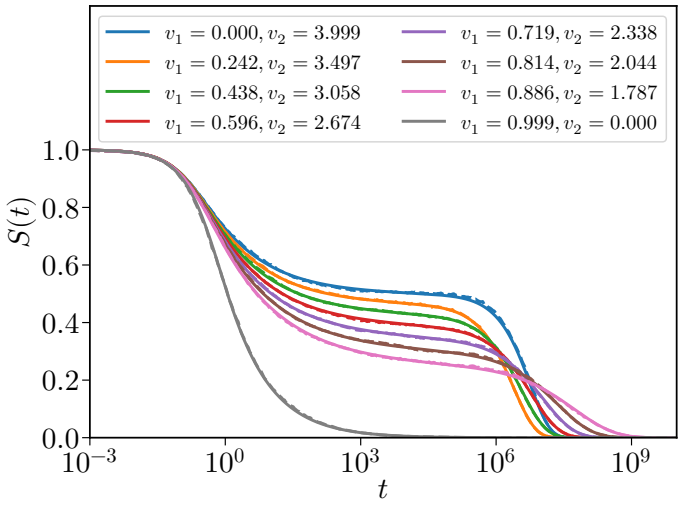
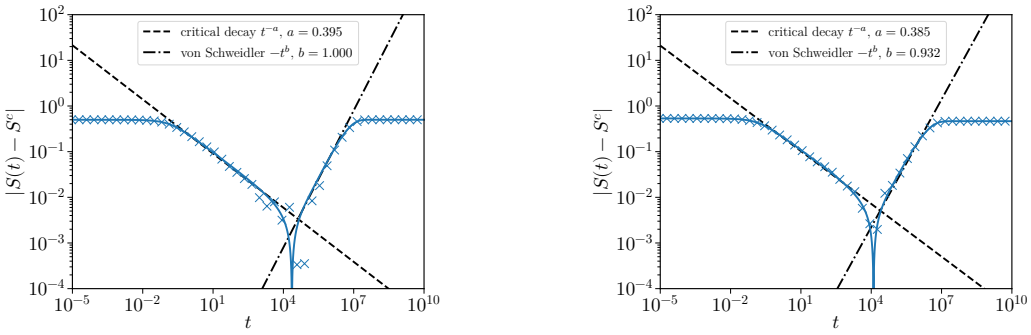


Figure 4.5: Solutions of the Mori–Zwanzig equation for the F_{12}-model near different critical points (solid lines), compared with correlation functions reconstructed by the neural-network-based inverse transform from their Laplace-domain representations obtained with the Filon method (dashed lines). The corresponding critical values are listed in tab. 4.2. The values shown here are obtained using $v_1 = v_1^c - \epsilon$, or $v_2 = v_2^c - \epsilon$ in the special case $v_1 = 0$, with $\epsilon = 10^{-3}$.

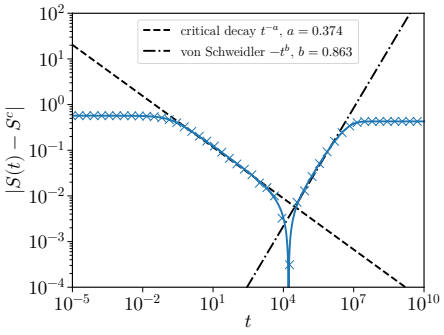
v_1^c	v_2^c
0.000000	4.000000
0.242887	3.497357
0.439480	3.057877
0.596621	2.673622
0.720224	2.337653
0.815400	2.043902
0.886558	1.787064
1.000000	0.000000

Table 4.2: Selection of critical parameter pairs (v_1^c, v_2^c) for the F_{12}-model, illustrating representative points along the bifurcation line separating the liquid and glassy regimes.

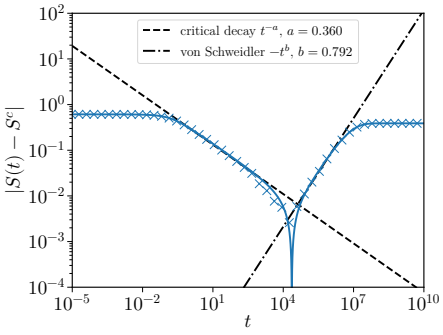


$v_1 = 0.0$, $v_2 = 3.999$

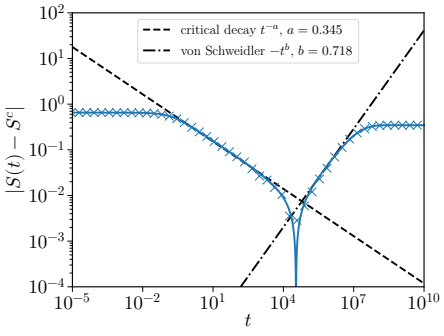
$v_1 = 0.242$, $v_2 = 3.497$



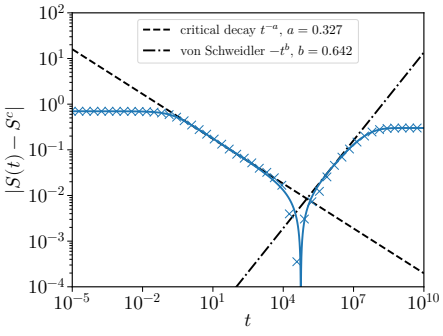
$v_1 = 0.438$, $v_2 = 3.058$



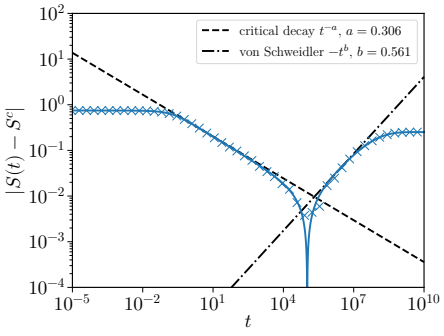
$v_1 = 0.596$, $v_2 = 2.674$



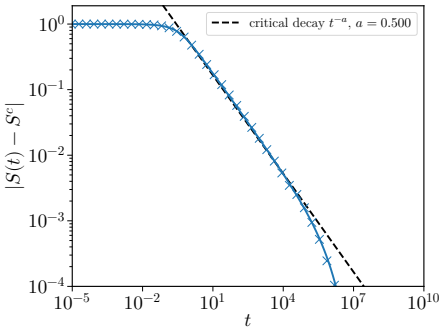
$v_1 = 0.719$, $v_2 = 2.338$



$v_1 = 0.814$, $v_2 = 2.044$



$v_1 = 0.886$, $v_2 = 1.787$



$v_1 = 0.999$, $v_2 = 0.000$

Figure 4.6: Asymptotic behavior of correlation functions for the F_{12}-model near different critical points. Solid lines indicate the solution of the Mori–Zwanzig equation, while crosses denote functions reconstructed by the NNLT from their Laplace-domain representations. Black lines indicate the asymptotic scaling laws.

Brownian Hard-Disk Model

The two-dimensional Brownian hard-disk model, detailed in sec. 4.1.5, provides a spatially resolved test for the NNLT due to its explicit wave-number dependence. In this context, the self-intermediate scattering function (SISF) is used as the benchmark correlation function for tagged-particle dynamics.

To validate the network, SISFs were calculated for a broad range of packing fractions ϕ below the critical point $\phi^c \approx 0.69057$. The NNLT successfully reconstructs the correlation functions for densities both close to and far from the glass transition, as illustrated in fig. 4.7. In particular, the network captures the emergence of long plateaus as $\phi \to \phi^c$, reflecting the slowing down of relaxation near dynamical arrest. This demonstrates its capability to resolve density-dependent changes in relaxation behavior.

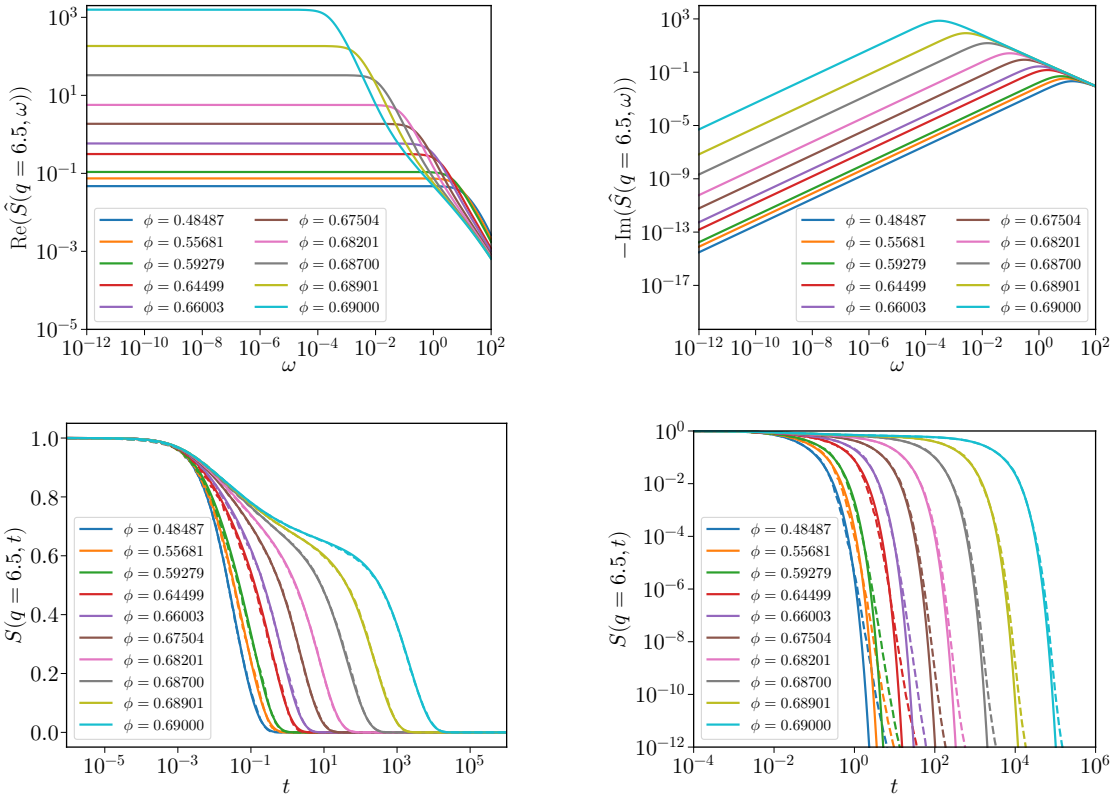


Figure 4.7: Results for the inverse Laplace transform of SISFs for the Brownian hard-disk model at wave number $q = 6.5$ for different packing fractions ϕ. The top panels display the Laplace-domain correlation functions obtained via the Filon method applied to numerical solutions of the Mori–Zwanzig equation. The bottom panels compare the corresponding time-domain correlation functions. Solid lines denote the numerical solutions, while dashed lines show the output of the NNLT.

Further validation is obtained by fixing the density and varying the wave number q, see fig. 4.8. As expected, larger q values probe shorter lengths and thus lead to faster decay of the SISF. Despite the significant differences in both time scales and functional shapes of the decay, the network consistently reconstructs the correlation functions across the entire probed q-range. This shows the ability of the NNLT to generalize to spatially structured correlation functions with non-trivial wave number dependence.

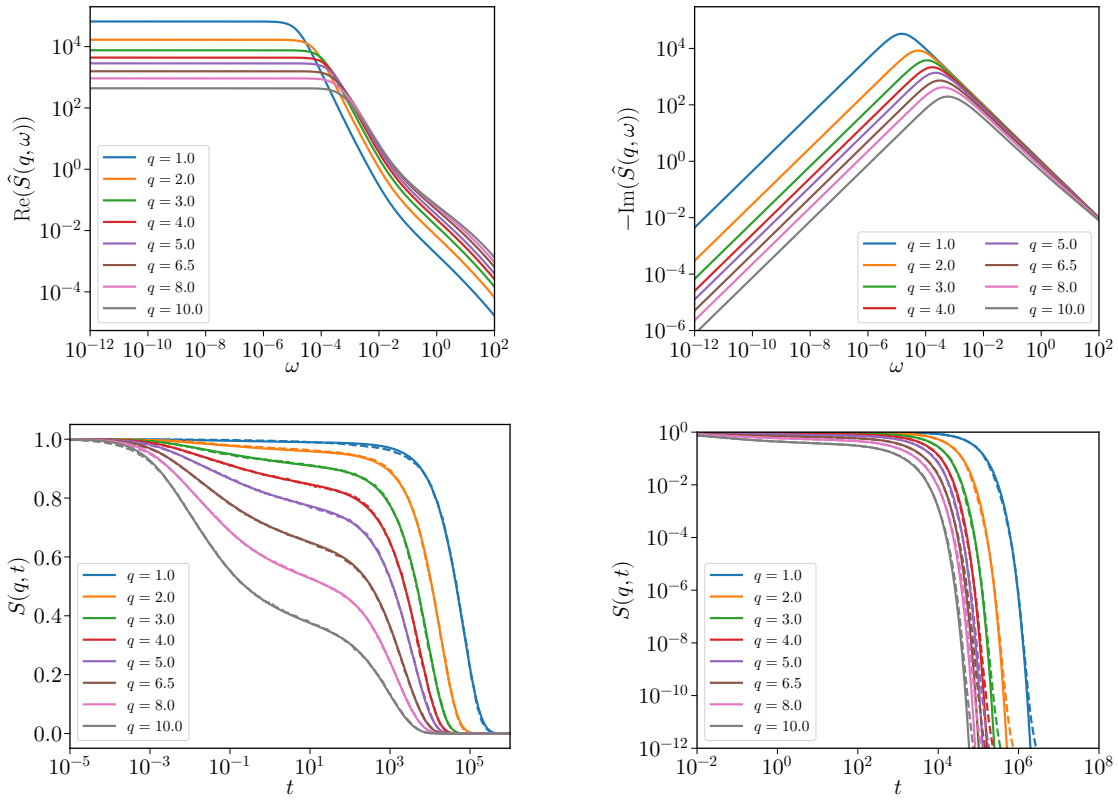


Figure 4.8: Results for the inverse Laplace transform of SISFs for the Brownian hard-disk model at packing fractions $\phi = 0.69$ for different wave numbers q. The top panels display the Laplace-domain correlation functions obtained via the Filon method applied to numerical solutions of the Mori–Zwanzig equation. The bottom panels compare the corresponding time-domain correlation functions. Solid lines denote the numerical solutions, while dashed lines show the output of the NNLT.

As in the schematic model, minor discrepancies appear in the long-time decay. These are likely due to finite machine precision, the numerical accuracy of the Filon method for the LT, or the truncation of the frequency spectrum, rather than shortcomings of the network itself. Overall, the results for the two-dimensional hard-disk model show that the NNLT generalizes well to realistic, spatially resolved correlation functions, extending its applicability beyond the synthetic exponential training set and schematic model.

Simulation Data of Brownian Particles

As a final validation step, the NNLT is applied to correlation functions obtained from Brownian dynamics simulations of two-dimensional hard-disks. The simulations are performed using the event-driven Brownian dynamics (EDBD) algorithm, described in sec. 2.1.2. The SISFs can be calculated from particle trajectories across a wide range of densities and wave numbers.

Despite employing ensemble averaging, the correlation functions naturally contain statistical noise on the order of 10^{-3}. Additional numerical artifacts arise from finite time steps, finite system sizes, and algorithmic approximations, though these primarily affect quantitative details rather than the qualitative relaxation behavior of the SISFs. To enhance the numerical stability of the Filon method, the correlation functions are truncated below a decay threshold of 10^{-3}. Beyond this cutoff, the long-time decay is smoothly extrapolated with a simple exponential of the form $A(t) = a \exp(-t/b)$, with parameters a and b determined from a least-squares fit to the last few data points. The number of fitted points is chosen adaptively based on the residual error of the fit, and the extrapolation is extended to values down to 10^{-6}. While this post-processing step is not strictly required for the NNLT, it substantially improves the quality of the Laplace domain representation by suppressing artifacts from noise.

The performance of the network is evaluated for different wave numbers q and across a broad range of packing fractions ϕ, spanning dilute and dense states gradually approaching dynamical arrest. Results for varying densities at fixed wave number are shown in fig. 4.9, while fig. 4.10 presents data for packing fraction $\varphi = 0.78$ at different wave numbers q. Although an ideal glass transition does not exist in simulations of finite systems [100, 101], this density is chosen because it exhibits pronounced slowing down of dynamics, while the correlation functions at the selected q-values still decay within accessible simulation times.

In both cases, the NNLT accurately reconstructs the time-domain SISFs from their Laplace-domain representations, yielding results closely resembling the original simulation data. Notably, the network also produces correct behavior at times beyond the range included in the Laplace transform by the Filon method, suggesting that it is capable of extrapolating long-time decay in a consistent manner.

Overall, the application to Brownian dynamics simulations confirms that the NNLT achieves performance comparable to that observed for MCT models. Even in the presence of moderate statistical noise and numerical artifacts, the method can reconstruct correlation functions, provided that noise-reduction measures such as tail extrapolation are applied.

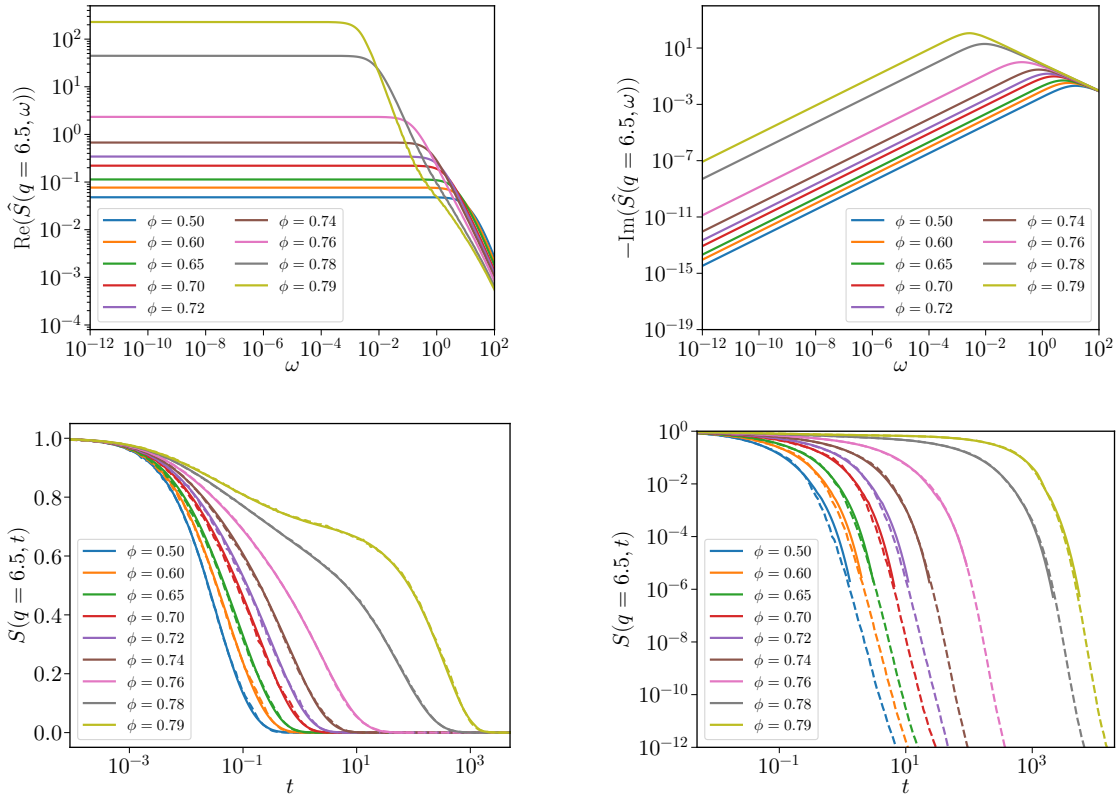


Figure 4.9: Results for the inverse Laplace transform of SISFs for Brownian dynamics at wave number $q = 6.5$ for different packing fractions ϕ. The top panels display the Laplace-domain SISFs obtained via the Filon method. The bottom panels compare the corresponding time-domain SISFs. Solid lines denote the simulation results, while dashed lines show the output of the NNLT.

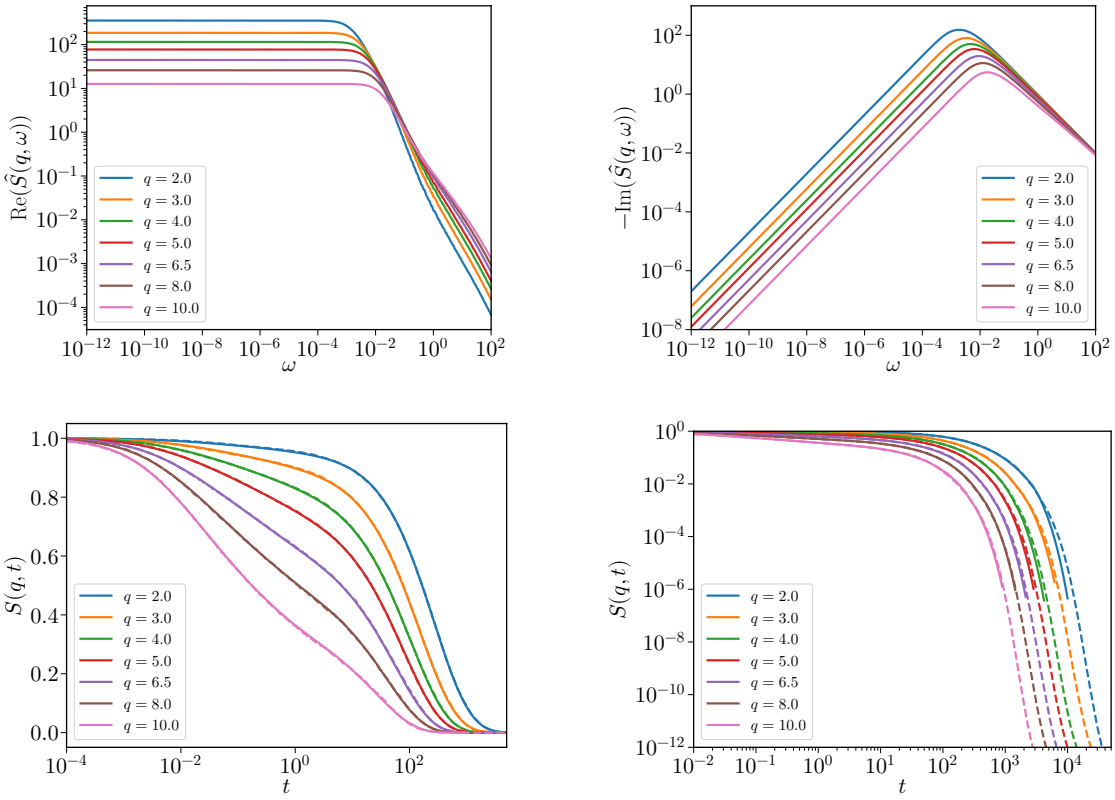


Figure 4.10: Results for the inverse Laplace transform of SISFs for Brownian dynamics at packing fraction $\phi = 0.78$ for different wave numbers q. The top panels display the Laplace-domain SISFs obtained via the Filon method. The bottom panels compare the corresponding time-domain SISFs. Solid lines denote the simulation results, while dashed lines show the output of the NNLT.

Summary

In this section, the neural-network-based inverse Laplace transform (NNLT) for correlation functions was introduced and validated. The architecture, data preparation, and training procedure were described, with particular emphasis on constructing a synthetic dataset of superpositions of exponential decays with known analytical Laplace transforms. Training on this family of functions enabled the network to accurately learn the mapping between Laplace domain input and time domain output across a broad range of relaxation timescales.

The performance of the trained network was validated with correlation functions obtained from both theory and simulation. The schematic F_{12} model, the wave number-dependent Brownian hard-disk model, and noisy simulation data from event-driven Brownian dynamics were considered. In all cases, the NNLT reliably reconstructed the original correlation functions from their Laplace-space representations obtained numerically by the Filon method. It successfully reproduced both short-time relaxation and long-time asymptotics, even in challenging scenarios such as two-step relaxation with extended plateaus near dynamical arrest or in the presence of statistical noise for simulation data. Importantly, the network demonstrated robustness against truncated frequency spectra, a situation in which conventional inverse transforms tend to fail.

The major advantage of the NNLT lies in its architecture, which enforces physically meaningful constraints such as the initial condition $S(q, 0) = 1$. This ensures stability and accuracy even when incomplete spectral information is provided, in contrast to direct inversion methods like the inverse Filon method, which lacks such constraints and therefore often fails to reproduce short-time behavior. The neural network approach is also able capture non-exponential behavior, including asymptotic scaling near critical points.

It should be emphasized, however, that the validation strategy employed here, i.e., transforming known correlation function to the Laplace domain with the Filon method and then applying the inverse transform, serves primarily as a proof of concept. The true strength of the approach emerges in applications where inversion pairs are unknown and direct inversion is either unstable or impossible. In particular, the NNLT enables the calculation of unknown memory kernels directly from measured or simulated correlation functions. Given a correlation function in the time domain, its LT can be computed and inserted into the Laplace-domain MZ equation. Applying the inverse transform to this memory kernel then gives its time-domain representation.

Having validated the method on both schematic and realistic data, the next step is to incorporate the NNLT into a framework to calculate memory kernels from correlation functions.

4.3 Neural Network Inverse Laplace Transform of the Memory Kernel

Through the Mori–Zwanzig equation in Laplace space, the memory kernel can be calculated from a given correlation function. In frequency representation, the Laplace transform (LT) of the memory kernel, $\hat{m}(q,\omega)$, is related to the LT of the correlation function, $\hat{S}(q,\omega)$, through

$$\hat{m}(q,\omega) = -\tau_0(q) - \frac{\hat{S}(q,\omega)}{\mathrm{i}\omega\hat{S}(q,\omega) - 1}, \tag{4.47}$$

where $\tau_0(q)$ denotes the microscopic relaxation time. For the self-intermediate scattering function (SISF) of Brownian particles, it is defined as

$$\tau_0(q) = \frac{1}{D_\mathrm{t} q^2}, \tag{4.48}$$

with D_t the translational diffusion coefficient. Eq. (4.47) is exact but numerically challenging, as inaccuracies in $\hat{S}(q,\omega)$, particularly at large ω, are strongly amplified in the evaluation of $\hat{m}(q,\omega)$.

To recover the time-domain memory kernel $m(q,t)$, an inverse LT of $\hat{m}(q,\omega)$ needs to be performed. Standard numerical inversion methods are notoriously unstable in this context, as truncated or noisy spectra lead to catastrophic cancellations, loss of high-frequency information, and unreliable reconstructions. To address these challenges, the neural-network-based inverse Laplace transform (NNLT), introduced in the previous section, is employed. The NNLT proved capable of approximating the inverse LT of correlation function even for noisy data.

One major complication arises because the NNLT is constrained to operate on normalized output ranges, i.e., [0, 1]. Memory kernels are not bounded and generally take values outside this interval. To make them compatible with the NNLT, a normalization scheme is introduced based on the initial value of the time-domain memory kernel,

$$m_0 = m(q, t = 0). \tag{4.49}$$

The network is then applied to the normalized memory kernel in Laplace space, $\tilde{m}(q,\omega) = \hat{m}(q,\omega)/m_0$. Provided that a reliable estimate of m_0 is available, this ensures reliable results across different wave numbers and densities.

If m_0 is chosen incorrectly, the inverse LT suffers from distortions due to the scaling

properties of the Laplace transform,

$$\mathcal{L}\{\alpha f(\alpha t)\} = \hat{f}(s/\alpha), \tag{4.50}$$

$$\mathcal{L}^{-1}\{\beta f(\beta s)\} = f(t/\beta). \tag{4.51}$$

An erroneous initial value therefore not only rescales the magnitude but also shifts the effective time scale of the calculated memory kernel. These errors are particularly severe at short times, where $m(q,t)$ is sensitive to high-frequency contributions that are difficult to resolve numerically. Fortunately, for solving the time-domain Mori–Zwanzig equation for a given memory kernel, the intermediate- and long-time behavior of $m(q,t)$ dominates.

Several strategies can be employed to estimate m_0:

- High-frequency asymptotics,

$$m_0 = \lim_{\omega\to\infty} \mathrm{i}\omega\, \hat{m}(q,\omega), \tag{4.52}$$

  which exploit the behavior of $\hat{m}$ at large ω but requires access to sufficiently high frequencies.

- Kramers–Kronig type relations,

$$m_0 = \frac{2}{\pi}\int_0^\infty \mathrm{Re}\left(\hat{m}(q,\omega)\right)\, \mathrm{d}\omega, \tag{4.53}$$

  which connect the initial value to the real part of the frequency-domain kernel.

- An iterative fitting scheme, where an initial guess for m_0 is made, using the above relations where possible. The normalized memory kernel $m^{\mathrm{guess}}(q,t)/m_0$ is then calculated with the NNLT. Its performance is evaluated by solving the time-domain Mori–Zwanzig equation, eq. (4.17) for ISFs or eq. (4.21) for SISFs, to obtain $S^{\mathrm{guess}}(q,t)$ and comparing it against the input correlation function. The procedure is iterated over multiple values close to the first guess until a sufficient intial value, m_0, is found.

The iterative scheme is the most robust, as it can compensate for poor analytical estimates or truncated spectral data.

Even with this refinement, the estimated initial values, m_0, may not perfectly satisfy known physical scaling laws. For example, $m_0(q) \propto 1/q^2$ is expected for small wave numbers in tagged particle dynamics [93, 102], as well as a strong increase with packing fraction ϕ. To enforce such constraints, the iterative scheme is applied across all available q-values at fixed ϕ, and the resulting $m_0(q)$ values are fit to the expected scaling forms. This procedure averages out noise to yield physically consistent scaling behavior.

Once the normalization scheme is established, the entire procedure can be validated against theoretical models, where the memory kernel is known. Successful validation demonstrates that the NNLT, combined with the m_0 estimation strategy, can robustly calculate memory kernels from Laplace-domain data. Subsequently, the method is applied to correlation function obtained in Brownian dynamics simulations, allowing to extract memory kernels directly from noisy correlation functions.

An overview of the workflow for calculating memory kernels from given correlation functions and validating the results is illustrated in fig. 4.11.

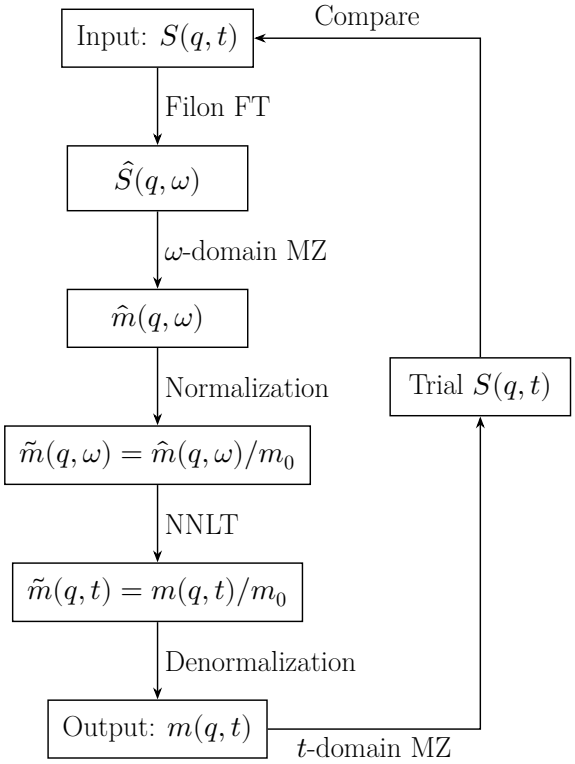


Figure 4.11: Sketch of the procedure for calculating the memory kernel $m(q,t)$ from a given correlation function $S(q,t)$. The workflow combines the Filon method for Fourier–Laplace transforms (sec. 2.3.2), the Laplace-domain Mori–Zwanzig equation (eq. (4.47)), normalization using an initial value estimate m_0, the neural-network-based inverse Laplace transform (sec. 4.2), and subsequent denormalization of the kernel. This memory kernel can then be inserted into the time-domain Mori–Zwanzig equation (eq. (4.17) for ISF or eq. (4.21) for SISF) to recover the corresponding correlation function $S(q,t)$, which is compared to the original input for validation.

4.3.1 Memory Kernel Calculation: Normalization by the Asymptotic Limit

To validate the general procedure for calculating memory kernels from correlation functions, the method is first applied to test cases where the initial value of the memory kernel, $m_0 = m(q, 0)$, is known exactly. In this context, the F_{12} schematic model and the Brownian hard-disk system provide ideal benchmarks. The memory kernels calculated via the NNLT can be compared directly to the corresponding solutions from MCT. The initial values are estimated from the asymptotic high-frequency behavior,

$$m_0 = \lim_{\omega \to \infty} \mathrm{i}\omega\, \hat{m}(q, \omega). \tag{4.54}$$

F_{12}-Model

Fig. 4.12 shows the memory kernels calculated for the F_1- and F_2-model, showing the NNLT successfully performs the inverse LT of the memory kernel. In these models, the asymptotic regime is well within the numerically stable frequency window, so no important contributions are lost through truncation. Additionally, the asymptotic estimate coincides with the actual initial value of the memory kernel. Deviations only appear in cases with extremely long plateaus, where the memory kernel approaches the temporal cutoff of the output-grid of the NNLT. Even in such cases, the large-time behavior is reproduced reasonably well. This can be attributed to the logarithmic error-weighting employed during network training, which emphasizes the accurate recovery of long-time dynamics, even when the absolute magnitude of function values becomes very small.

To assess whether these memory kernels actually produce the correct correlations functions, a self-consistency check is performed by solving the time domain Mori–Zwanzig equation providing the calculated memory kernel and comparing the resulting SISF to the input correlation function, as shown in fig. 4.13. The excellent agreement confirms that the method can produce the expected output shape and also contain the dynamical structure necessary to recover the input correlation function. This demonstrates the applicability of the approach, at least when m_0 can be accurately estimated from the Laplace-domain data.

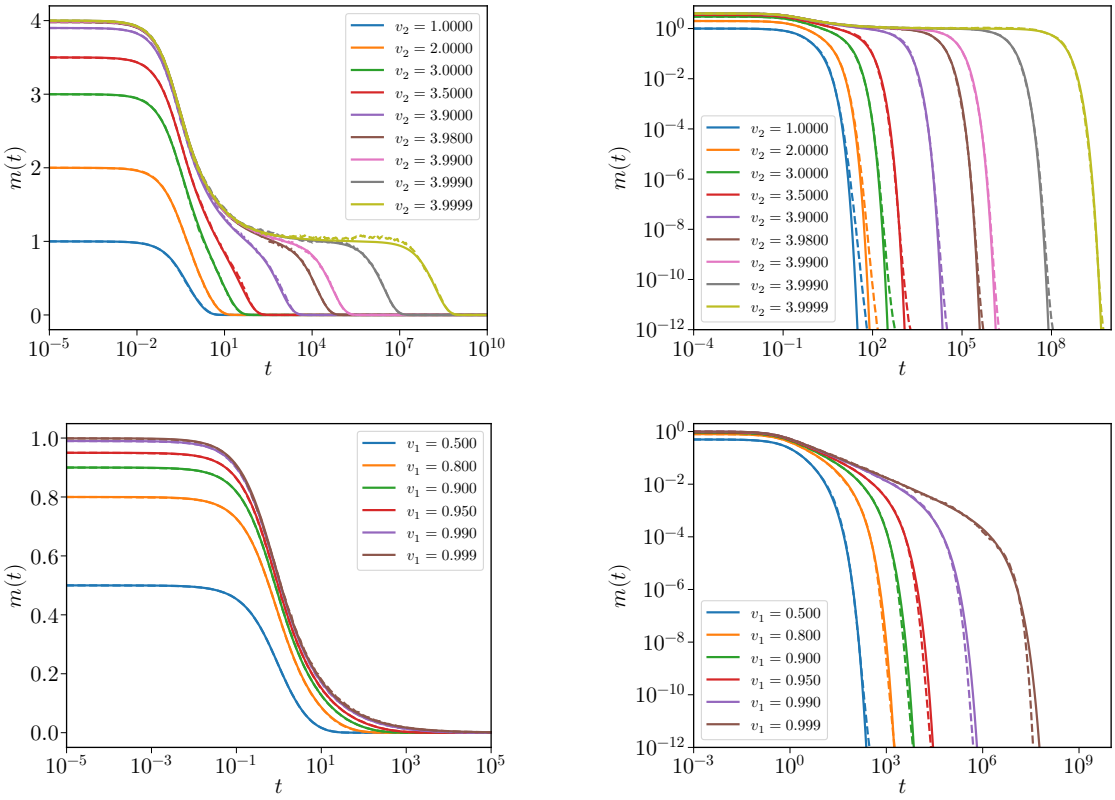


Figure 4.12: Memory kernels calculated from correlation functions using the Laplace-domain Mori–Zwanzig equation and the NNLT with normalization by the asymptotic limit (dashed), compared to the numerical mode coupling theory results (solid) for the F_{12}-model. Top panels show the F_2-model, bottom panels the F_1-model.

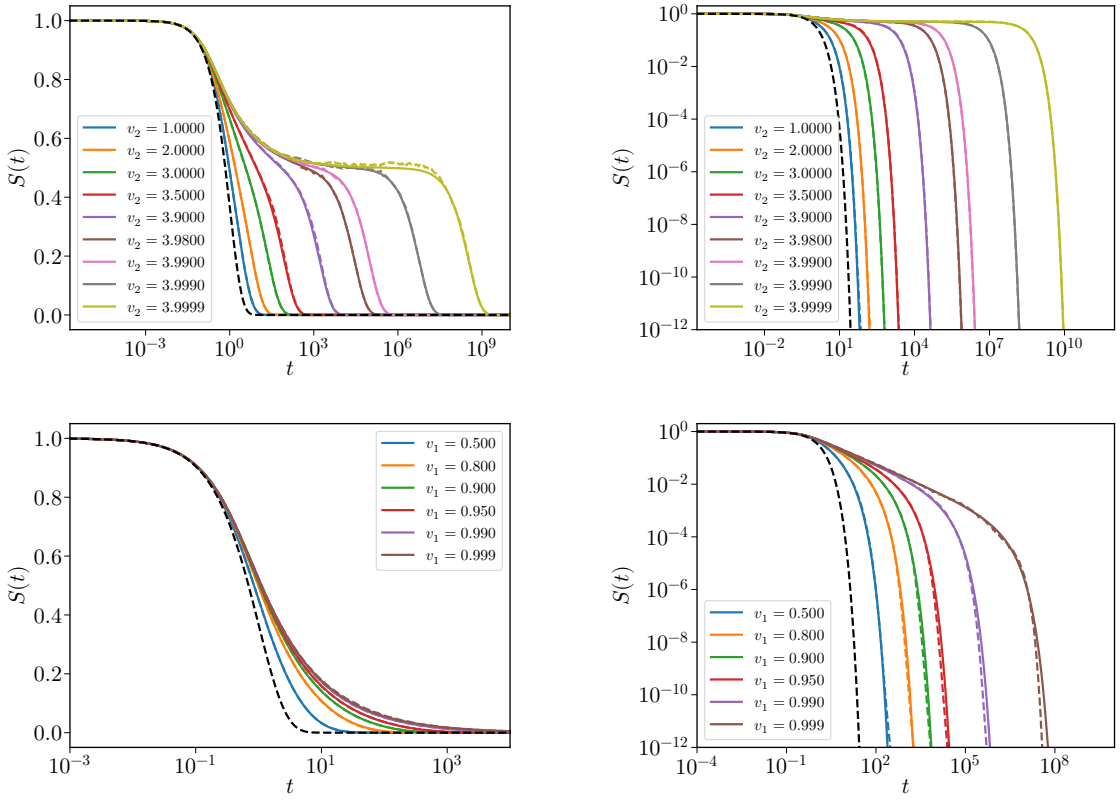


Figure 4.13: Correlation functions obtained from the solution of the time-domain Mori–Zwanzig equation for the F_{12}-model. Solid lines show the mode coupling theory results, while dashed lines correspond to solutions using the calculated memory kernels from fig. 4.12. Top panels show the F_2-model, bottom panels the F_1-model.

Brownian Hard-Disk Model

The second model highlights the limitations of the asymptotic normalization method. For the two-dimensional Brownian hard-disk model, the initial value $m_0 = m(q, 0)$ is known exactly, but the high-frequency asymptotic estimate does not coincide with the actual value. Additionally, truncation errors and numerical noise render the limit unreliable for normalization.

The consequences are demonstrated in fig. 4.14, which compares two normalizations. One uses the exact normalization constant, the other the asymptotic estimate. The impact of incorrect normalization is clearly visible in the short-time behavior, where the denormalized memory kernel deviates strongly. This is expected, since the neural network outputs are bounded within [0, 1], and rescaling with an incorrect m_0 propagates directly into the resulting memory kernel. Interestingly, the differences diminish at intermediate and long times, and both memory kernels converge to nearly identical results. This insensitivity at long times can be explained by the properties of the LT and the training procedure of the neural network, with the latter emphasizing the reproduction of features across the intermediate- and long-time scales rather than exact short-time behavior.

The consequences of the discrepancies between memory kernels for the different initial value estimates are examined by comparing the resulting SISFs obtained from the time domain Mori–Zwanzig equation, see fig. 4.15. Despite the substantial deviations of $m(q, t)$ at short times, the correlation functions remain remarkably similar, with the SISFs obtained from the memory kernel calculated with correct normalization showing almost perfect agreement with the theory. The characteristic relaxation time scales are preserved, with only minor shifts in amplitude and time constants. This demonstrates that the integrative nature of the Mori–Zwanzig equation tends to smooth out short-time errors in the memory kernel.

These observations shows both strengths and limitations of the asymptotic normalization strategy combined with the Laplace-domain MZ equation and the NNLT. On the one hand, it confirms that the long-time behavior of correlation functions can be reproduced faithfully even when the short-time normalization of the memory kernel is not exact. On the other hand, it shows the degeneracy of the inversion problem. Different initial value estimates may yield memory kernels that differ at short times but still reproduce reasonably correct long-time behavior for $S(q, t)$. While this may be beneficial, it also implies that validation of memory kernels requires more than consistency checks with the provided correlation function. In particular, analyzing the scaling laws of $m_0(q)$ across wave numbers and densities is essential to ensure physical reliability.

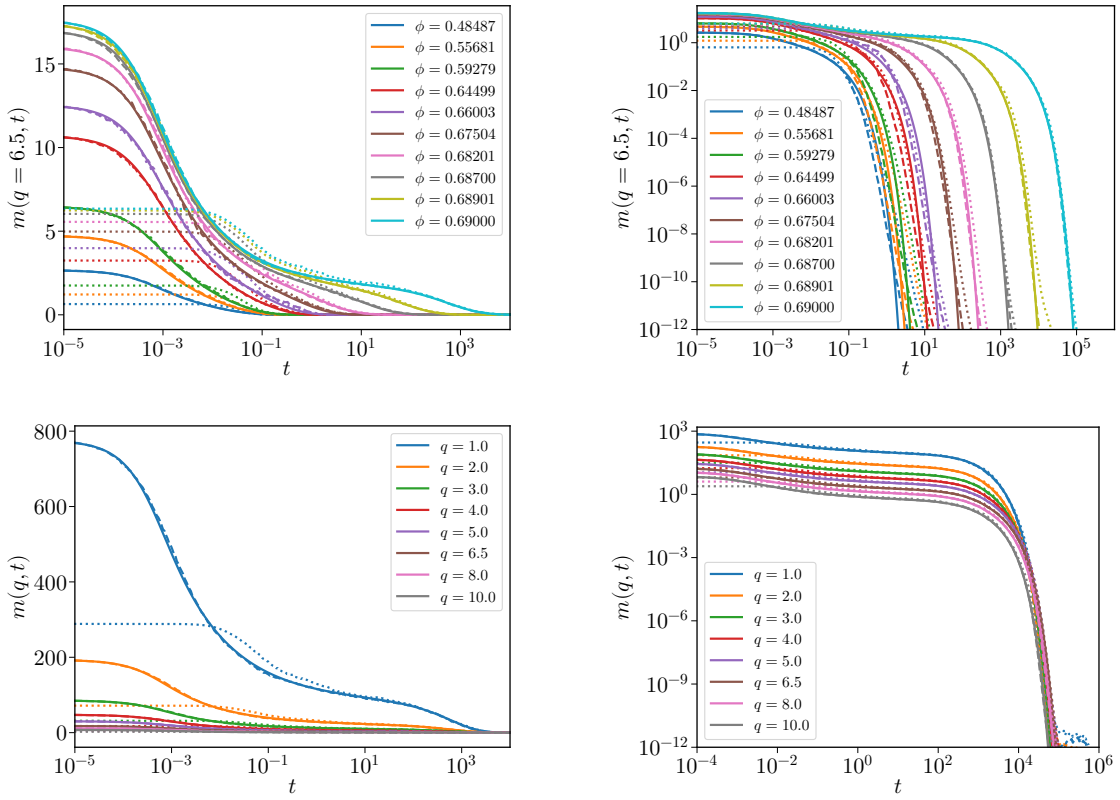


Figure 4.14: memory kernels calculated from the self-intermediate scattering functions using the Laplace-domain Mori–Zwanzig equation and the NNLT with correct normalization (dashed) and asymptotic-limit normalization (dotted), compared to the mode coupling theory results (solid) for Brownian hard-disks. Top panels show different packing fractions ϕ at wave number $q = 6.5$, bottom panels show different wave numbers q at packing fraction $\phi = 0.69$.

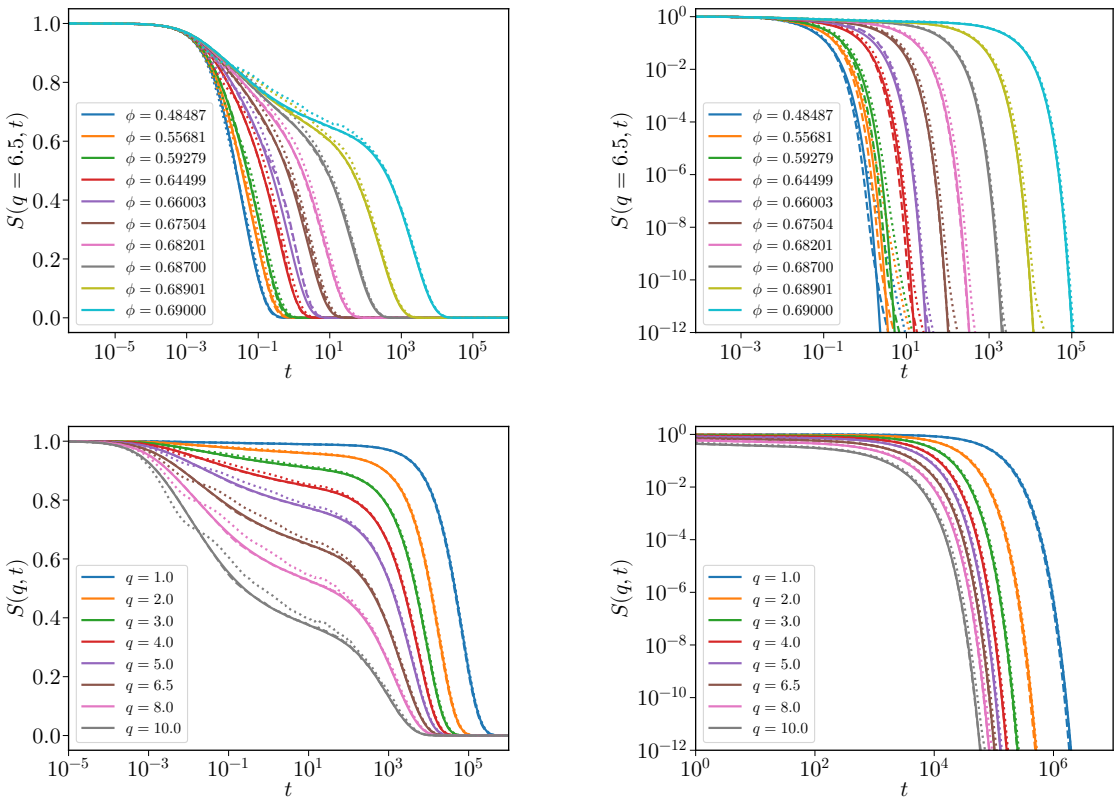


Figure 4.15: SISFs obtained from the solution of the time-domain Mori–Zwanzig equation for Brownian hard-disks. Solid lines show results from mode coupling theory. Solutions using the calculated memory kernels from fig. 4.14 are shown for correct normalization (dashed) and for normalization by the asymptotic-limit estimate (dotted). Top panels show different packing fractions ϕ at wave number $q = 6.5$, bottom panels show different wave numbers q at packing fraction $\phi = 0.69$.

4.3.2 Memory Kernel Calculation: Normalization by an Iterative Fitting Scheme

The calculation of memory kernels from the self-intermediate scattering function is highly successful when the correct initial value, $m_0 = m(q, 0)$, is supplied. However, estimating m_0 from the high-frequency asymptotic behavior becomes unreliable whenever large portions of the spectrum lie outside the numerically stable frequency window. This limitation is already encountered in the Brownian hard-disk MCT model and becomes even more pronounced for simulation data, where statistical noise further restricts the accessiblility of the asymptotic limit.

The maximum usable frequency of the NNLT is ultimately given by a combination of multiple factors, dominantely by the noise level of the input data, the numerical stability of the Filon method used for the Fourier transform, and the behavior of the Laplace-domain Mori–Zwanzig equation to calculate the memory kernel. When the high-frequency regime cannot be resolved with sufficient accuracy, an alternative strategy must be adopted to estimate m_0. To this end, an iterative normalization scheme is proposed in which m_0 is treated as a variational parameter.

Procedure

The iterative normalization scheme is implemented as follows:

- Select a trial value for m_0.
- Compute the candidate memory kernel $m(q, t)$ using the Laplace domain Mori–Zwanzig equation, the NNLT, and subsequent denormalization.
- Solve the time domain Mori–Zwanzig equation with this candidate memory kernel and calculate the corresponding SISF.
- Compare the reconstructed SISF with the input correlation function $S(q, t)$.
- Repeat the procedure for a range of trial values of m_0 and minimize the error between predicted and input SISF to identify the optimal normalization constant.

This empirical approach provides a reliable estimate of m_0 even when the asymptotic methods fail, requiring no additional assumptions beyond providing a range of trial values. The Brownian hard-disk model, where m_0 is exactly known, serves as a crucial benchmark for validating the scheme. Moreover, due to its qualitative similarity to results from Brownian particle simulations, it serves as a meaningful proxy for assessing the robustness of the suggested method when applied to noisy numerical data.

Convergence Properties

Fig. 4.16 illustrates the influence of different initial values on the calculated memory kernels. While the short-time behavior is strongly affected by the choice of m_0, mostly owing to the denormalization of the normalized output of the NNLT, the memory kernels converge after the initial decay and exhibit remarkably similar intermediate-time behavior. This convergence reflects the time- and frequency-scaling properties of the LT.

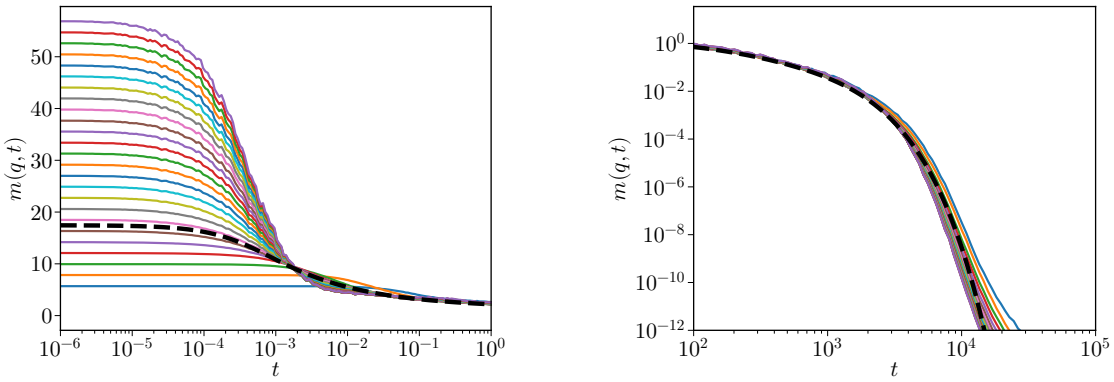


Figure 4.16: Memory kernels calculated from the Laplace-domain Mori–Zwanzig equation combined with the NNLT and normalization for different initial value estimates m_0 in the Brownian hard-disk model at packing fraction $\phi = 0.68930$ and wave number $q = 6.5$. The dashed black line shows the reference memory kernel from mode coupling theory.

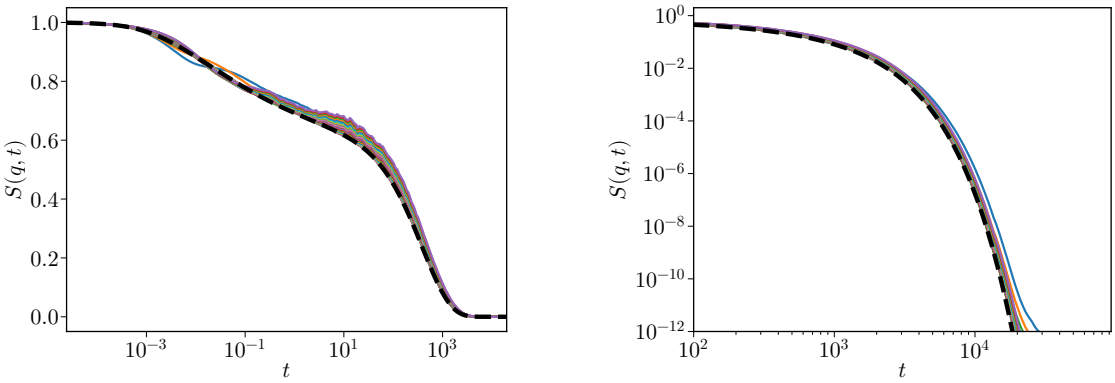


Figure 4.17: SISFs obtained by solving the time-domain Mori–Zwanzig equation with the memory kernels shown in fig. 4.16 for different initial value estimates m_0. The dashed black line shows the reference SISF obtained from the hard-disk memory kernel closure. Shown is packing fraction $\phi = 0.68930$ at wave number $q = 6.5$

When these memory kernels are used to solve the time domain Mori–Zwanzig equation, the resulting SISFs remain close to the ground-truth solution. As shown in fig. 4.17, deviations are most apparent in the intermediate plateau region, while the overall relaxation behavior is well reproduced. This insensitivity indicates that small inaccuracies in m_0 are not

catastrophic, as the dynamics are dominated by the intermediate- and long-time values of the memory kernel.

Self-Consistency Metric

To quantify the agreement between the input and reconstructed SISFs, the mean squared error and a normalized overlap integral are considered as error metrics. Since both yield comparable results, the overlap integral is used in the following.

Since the initial value is expected to scale as $m_0 \sim 1/q^2$ for small wave numbers [93, 102], the self-consistency error is plotted as a function of $q^2 m_0$, see fig. 4.18. The errors reveal relatively broad minima, indicating that a wide range of initial values produce similar results. In some cases, the local minima split, suggesting interfering effects between the Laplace-domain MZ calculation and the inverse transform performed by the NNLT. Unfortunately, the exact reason why the minima only sometimes split remains unclear.

Although the iterative scheme may identify memory kernels that reproduce the SISFs well, the optimal m_0 values do not always match the known ones, particularly at larger wave numbers q. This discrepancy likely reflects the accumulation of multiple error sources, including numerical noise, network imperfections, and truncation of the frequency spectrum, in combination with a certain insensitivity due to the degree of degeneracy of the memory kernel with respect to the initial value.

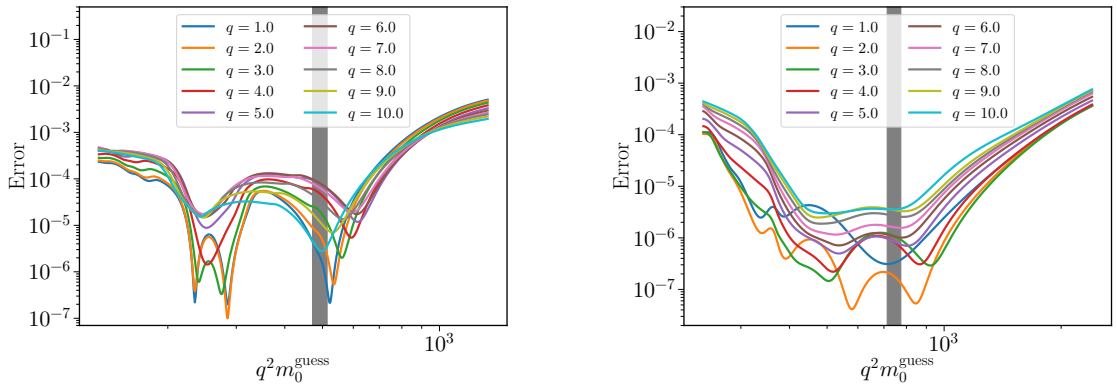


Figure 4.18: Overlap integral error between the SISFs of the Brownian hard-disk model and those calculated from the time-domain Mori–Zwanzig equation using memory kernels obtained via the Laplace-domain Mori–Zwanzig equation combined with the NNLT and normalization by different initial value estimates m_0, shown as a function of $q^2 m_0$. Shaded areas indicate the range of $q^2 m_0$ values from MCT. The left panel depicts packing fraction $\phi = 0.65246$, while the right shows $\phi = 0.68930$.

Scaling Correction

While optimizing m_0 independently for each wave number may yield the best individual memory kernels, it fails to reproduce the expected scaling laws with q. To restore physical consistency, parametric scaling is introduced,

$$m_0(q) = \frac{a}{q^2}\left(1 - bq\right), \tag{4.55}$$

with a and $b > 0$ determined by minimizing the average self-consistency error across all available q-values at fixed packing fraction. This procedure effectively averages out noise and numerical artifacts, enforcing consistent scaling across wave numbers and, by extension, across packing fractions ϕ, where the initial value of memory kernels is expected to increase drastically with ϕ.

The fitted q-dependence is shown in fig. 4.19, while the resulting packing fraction dependence is depicted in fig. 4.20. Although some variation is observed in the parameter a, the scaling behavior of m_0 is improved significantly, particularly at high densities where deviations are largest. An overview of the errors across densities is provided in tab. 4.3. While the RMSE roughly doubles compared to optimizing m_0 individually, it remains within an acceptable range, justifying the trade-off to obtain physically consistent scaling.

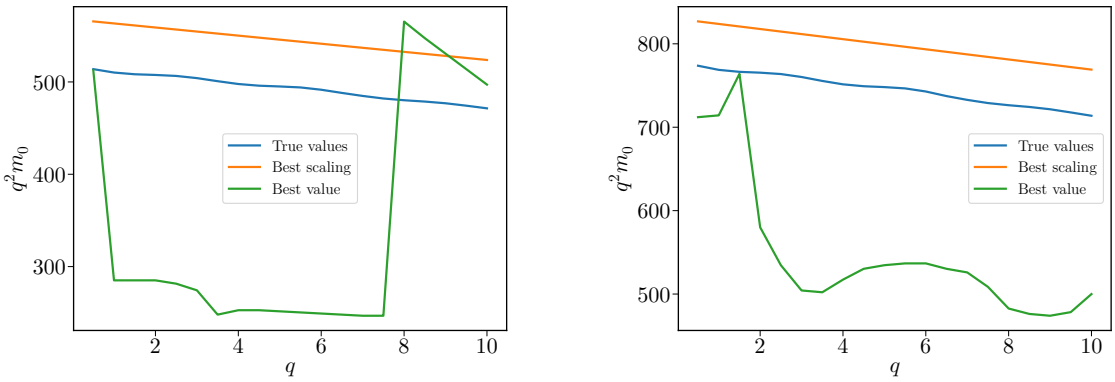


Figure 4.19: Best-fit scaling behavior of the initial value $m_0(q)$ as a function of wave number q for packing fractions $\phi = 0.65246$ (left) and $\phi = 0.68930$ (right), compared to the exact scaling from mode coupling theory and to initial values obtained by independently minimizing the error for each wave number.

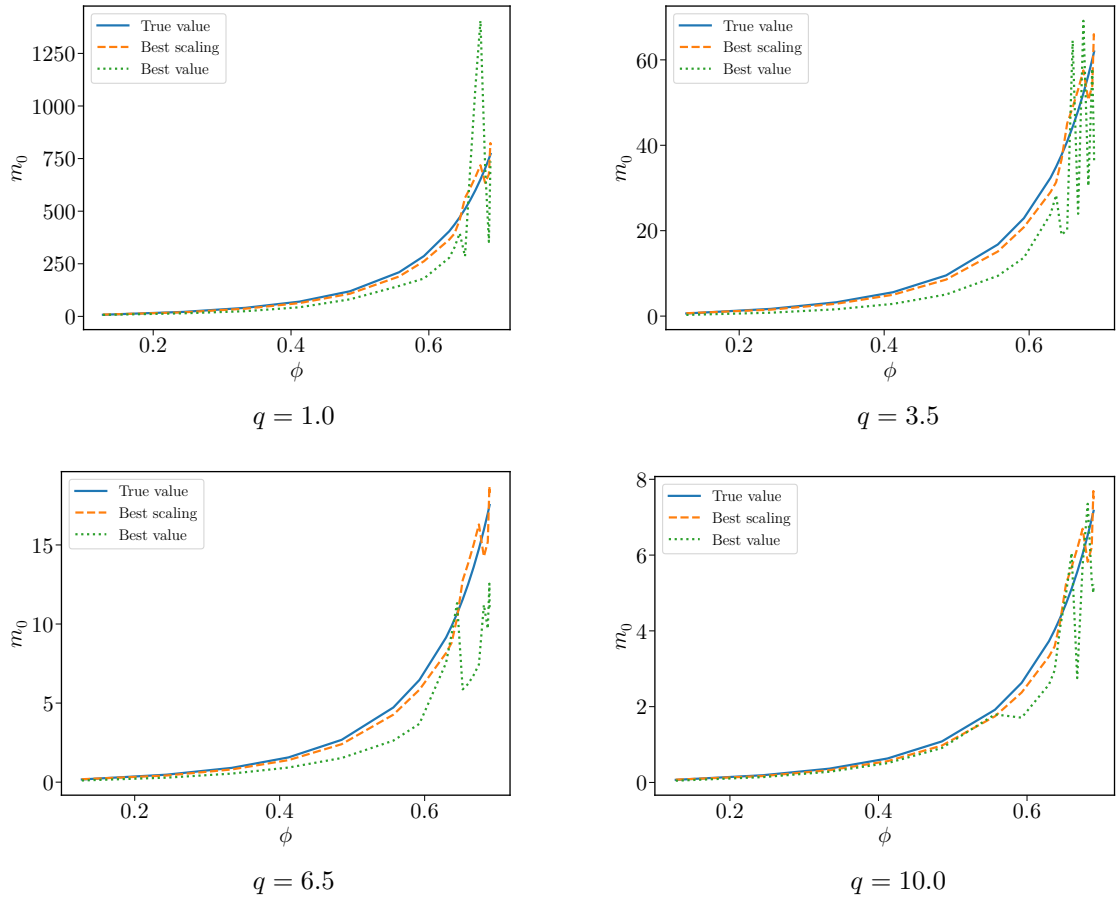


Figure 4.20: Best-fit scaling behavior of the initial value $m_0(\phi)$ as a function of packing-fraction ϕ for different wave numbers q, compared to the exact scaling from mode coupling theory and to initial values obtained by independently minimizing the error for each wave number.

ϕ	$\mathrm{RMSE}_{\mathrm{scal}}$	$\mathrm{RMSE}_{\mathrm{best}}$
0.12707	$1.542 \cdot 10^{-3}$	$5.305 \cdot 10^{-4}$
0.24397	$6.751 \cdot 10^{-4}$	$1.606 \cdot 10^{-4}$
0.33389	$7.768 \cdot 10^{-4}$	$1.840 \cdot 10^{-4}$
0.41225	$7.658 \cdot 10^{-4}$	$2.373 \cdot 10^{-4}$
0.48547	$8.986 \cdot 10^{-4}$	$3.528 \cdot 10^{-4}$
0.55681	$1.382 \cdot 10^{-3}$	$5.932 \cdot 10^{-4}$
0.59279	$1.991 \cdot 10^{-3}$	$7.532 \cdot 10^{-4}$
0.62951	$3.318 \cdot 10^{-3}$	$8.902 \cdot 10^{-4}$
0.63721	$2.680 \cdot 10^{-3}$	$9.892 \cdot 10^{-4}$

ϕ	$\mathrm{RMSE}_{\mathrm{scal}}$	$\mathrm{RMSE}_{\mathrm{best}}$
0.64473	$2.355 \cdot 10^{-3}$	$9.766 \cdot 10^{-4}$
0.65246	$3.299 \cdot 10^{-3}$	$1.998 \cdot 10^{-3}$
0.66003	$8.563 \cdot 10^{-3}$	$4.264 \cdot 10^{-3}$
0.66756	$1.039 \cdot 10^{-2}$	$3.623 \cdot 10^{-3}$
0.67487	$4.218 \cdot 10^{-3}$	$3.695 \cdot 10^{-3}$
0.68171	$1.646 \cdot 10^{-3}$	$1.170 \cdot 10^{-3}$
0.68711	$1.593 \cdot 10^{-3}$	$9.987 \cdot 10^{-4}$
0.68930	$1.077 \cdot 10^{-3}$	$9.141 \cdot 10^{-4}$
0.68966	$1.571 \cdot 10^{-3}$	$1.269 \cdot 10^{-3}$

	$\mathrm{RMSE}_{\mathrm{scal}}$	$\mathrm{RMSE}_{\mathrm{best}}$
Overall	$3.752 \cdot 10^{-3}$	$1.794 \cdot 10^{-3}$

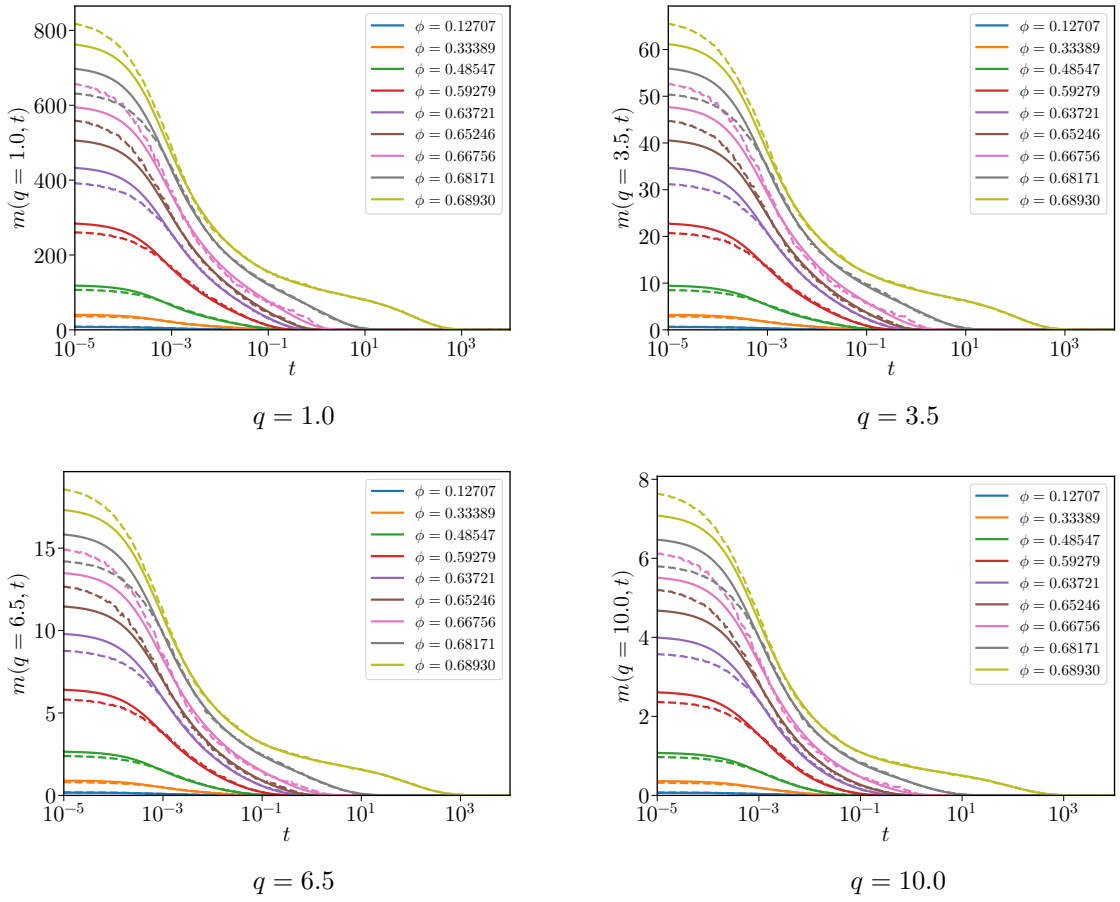
Table 4.3: RMSE between the SISFs of the Brownian hard-disk model and those calculated from the time-domain Mori–Zwanzig equation using memory kernels obtained via the Laplace-domain Mori–Zwanzig equation combined with the NNLT. Results are shown for normalization with initial values from the best-fit q-scaling behavior and from the individually optimized initial values at each wave number, across different packing fractions ϕ.

Results

The memory kernels obtained with initial values from scaling-based fits are shown in fig. 4.21, with the corresponding SISFs following from the solution of the time-domain MZ equation depicted in fig. 4.22. With the exception of one outlier at $\phi = 0.66756$, the results coincide with the MCT results. Importantly, the density scaling of m_0 is now reproduced well across the full range. The deviations at very high densities may result from the onset of glassy dynamics near the transition point.

For comparison, memory kernels and corresponding SISFs obtained from initial values optimized for each (ϕ, q) correlation function individually are presented in fig. 4.23 and fig. 4.24, respectively.

Overall, the iterative scheme provides a robust and practical method for estimating m_0 where asymptotic methods fail. While it may not perfectly recover the correct initial values, it produces memory kernels that can reproduce correlation functions reliably, especially when explicitly enforcing physically motivated scaling laws. This makes the method particularly suitable for application to simulation data, where noise is present and direct knowledge of m_0 or the memory kernel is unavailable.



$q = 1.0$ $q = 3.5$

$q = 6.5$ $q = 10.0$

Figure 4.21: Memory kernels obtained from the Laplace-domain Mori–Zwanzig equation combined with the NNLT and normalization using the best-fit scaling behavior for the initial value $m_0(q)$ (dashed), compared to the MCT memory kernels (solid) for different packing fractions ϕ at fixed wave numbers q.

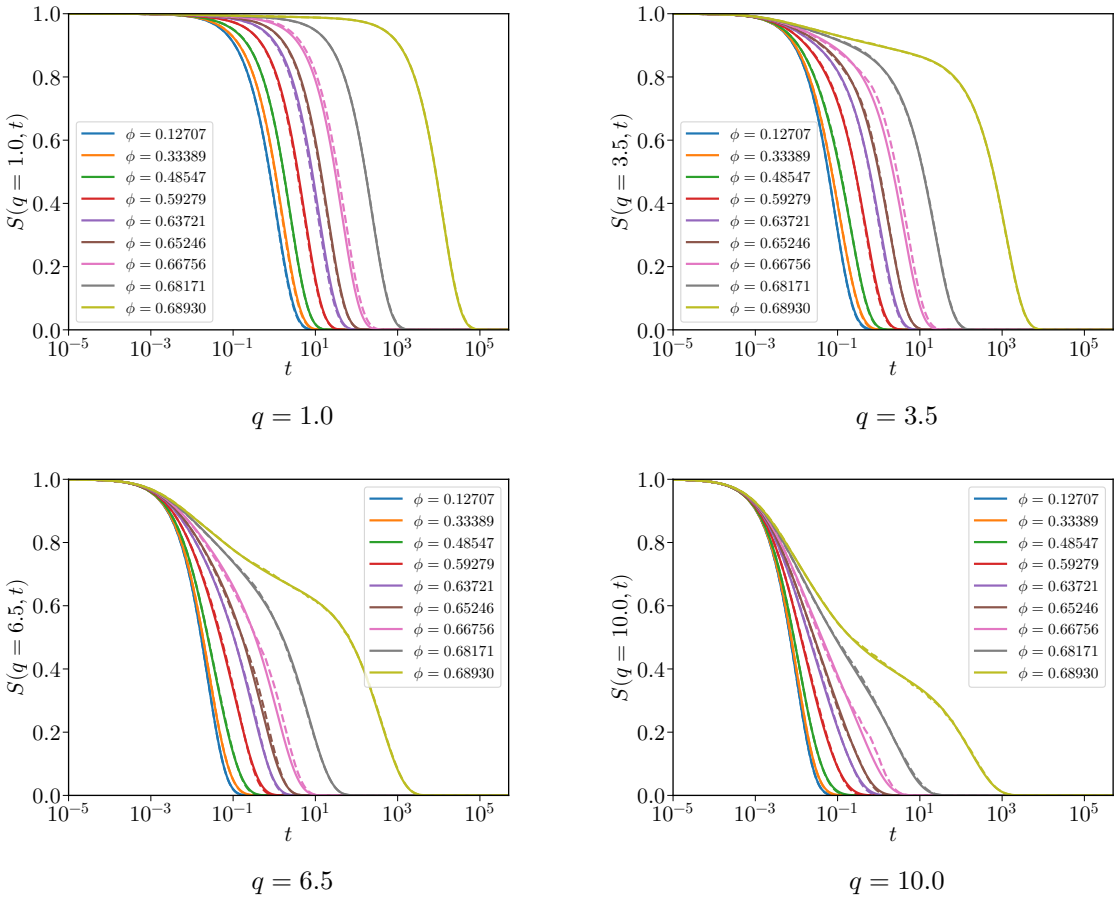
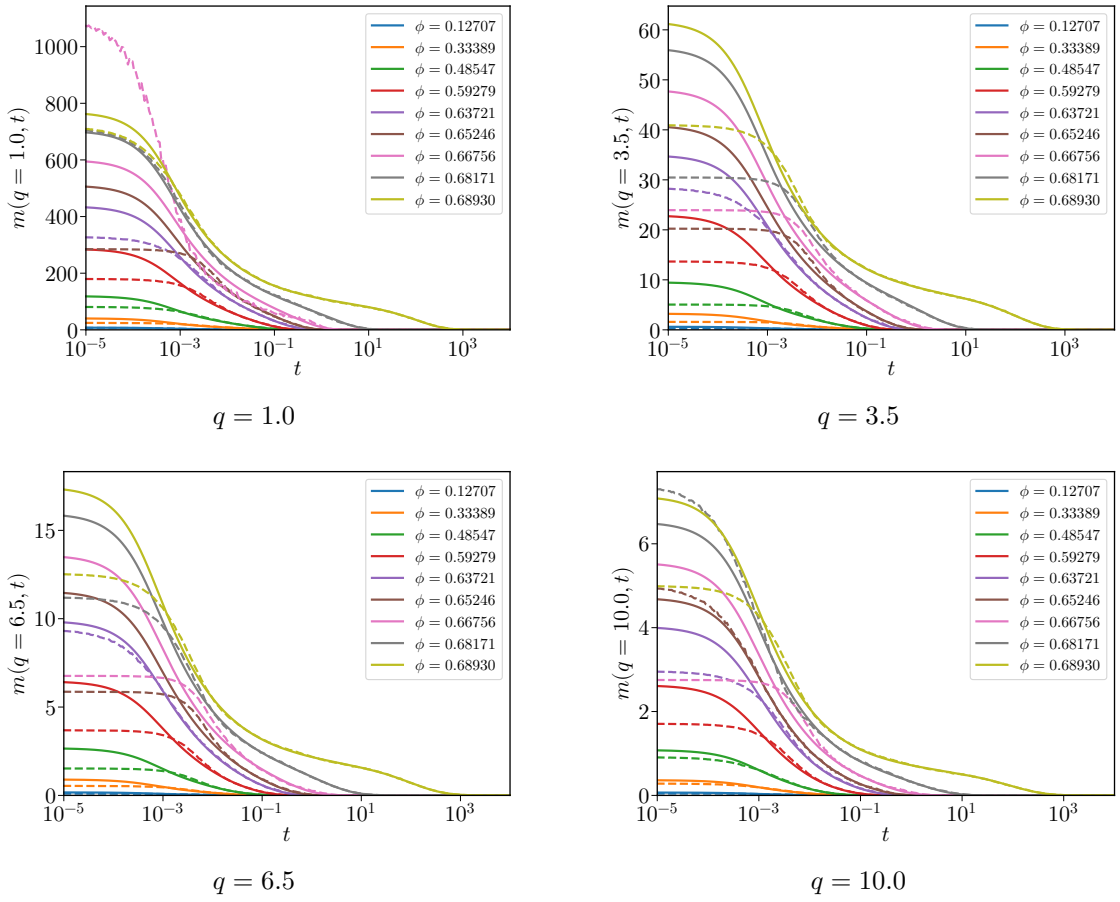


Figure 4.22: SISFs obtained from the solution of the time-domain Mori–Zwanzig equation with memory kernels obtained with normalization using best-fit scaling behavior for the initial value $m_0(q)$ (dashed, see fig. 4.21), compared to the mode coupling theory SISFs (solid) for different packing fractions ϕ at fixed wave numbers q.



$q = 1.0$ $q = 3.5$

$q = 6.5$ $q = 10.0$

Figure 4.23: Memory kernels obtained from the Laplace-domain Mori–Zwanzig equation combined with the NNLT and normalization using the smallest individual error initial value estimate m_0 (dashed), compared to the MCT memory kernels (solid) for different packing fractions ϕ at fixed wave numbers q.

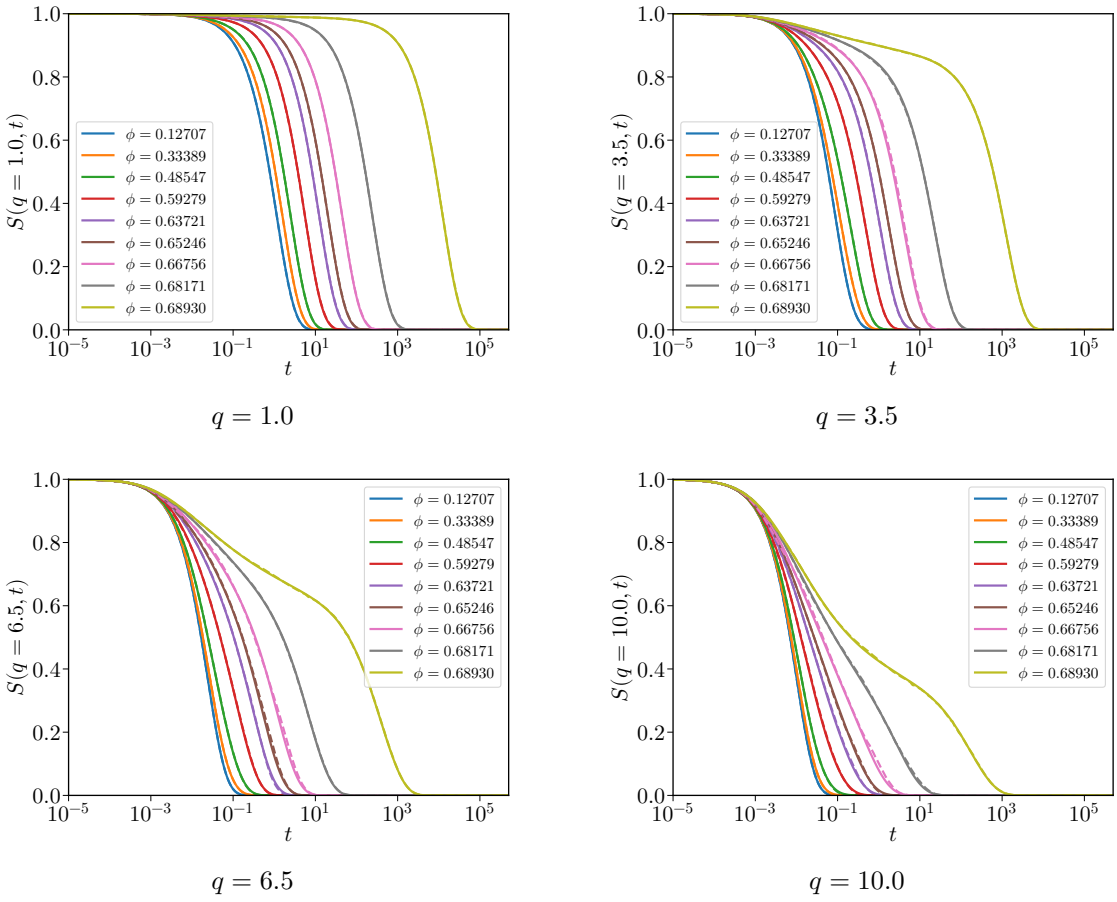


Figure 4.24: SISFs obtained from the solution of the time-domain Mori–Zwanzig equation with memory kernels obtained with the smallest individual error initial value estimate m_0 (dashed, see fig. 4.23), compared to the mode coupling theory SISFs (solid) for different packing fractions ϕ at fixed wave numbers q.

Summary

In this section, the neural network inverse Laplace transform (NNLT) of correlation functions was used in combination with the Laplace-domain Mori–Zwanzig equation to calculate memory kernels from correlation functions.

A major difficulty arised from the fact that memory kernels are unnormalized, whereas the NNLT is only designed to operate on normalized functions. To overcome this, a normalization scheme based on the initial value $m_0 = m(q, 0)$ was developed. When m_0 can be reliably estimated from asymptotic high-frequency behavior, or is known a priori, the method yields remarkably accurate memory kernels. However, in cases where the asymptotic regime lies outside the numerically accessible frequency window, such as in the Brownian hard-disk model or in noisy simulation data, direct estimates of m_0 become unreliable. To address this challenge, an iterative fitting scheme was proposed, treating m_0 as a variational parameter. This procedure proved robust even for truncated and noisy spectra, a regime where conventional inversion techniques, e.g., inverse Filon method, typically fail due to numerical instabilities.

The method was validated against the Brownian hard-disk model, where memory kernels are available from MCT. These tests showed that the method faithfully calculates memory kernels, while also revealing the relative insensitivity of long-time dynamics to errors in the short-time normalization. At the same time, the importance of correct scaling laws for m_0 with wave number became apparent. To enforce scaling laws, the fitting scheme for the initial value was extended based on hydrodynamic considerations.

Having established and validated the methodology, the stage is now set to apply the framework to simulation data, where the true memory kernel are not known. This presents a significant step forward, as it enables direct access to memory kernels and thereby allows stringent comparison between simulation results and theoretical predictions of MCT. The following section will employ this approach to calculate memory kernels for Brownian systems.

4.4 Memory Kernel Calculation for Brownian Dynamics Simulations

Having established a systematic procedure to calculate memory kernels in the previous section, it is now applied to actual simulation data. Specifically, particle trajectories obtained from simulations of two-dimensional Brownian hard-disk systems over a broad range of packing fractions ϕ are analyzed. The simulations are performed using the event-driven Brownian dynamics algorithm [42, 43], see sec. 2.1.2 for details. The simulation code has been extensively tested and is known to yield reliable static and dynamic observables.

For each state point, 200 independent ensembles are generated, each simulated for $2 \cdot 10^9$ integration steps with a timestep of $\Delta t = 10^{-5}$. Assuming ergodicity, trajectories are additionally divided into 50 sub-ensembles with linearly spaced starting points, effectively increasing the available statistics without additional computational cost.

From the particle trajectories, the self-intermediate scattering function (SISF) is computed. These SISFs are subject to statistical noise, which scales approximately as $1/\sqrt{N_{\text{part}} N_{\text{ens}}}$, where N_{part} denotes the number of particles and N_{ens} the number of (independent) ensembles.

To further reduce noise and ensure high-quality input data, the self-intermediate scattering functions (SISFs) requires additional preprocessing. Once $S(q,t)$ decays below a specified noise level, it is truncated, and the tail is extrapolated by fitting a simple exponential function to the last $4-6$ data points, depending on fit quality. This preprocessing avoids numerical instabilities associated with integrating over noise amplitudes of order 10^{-3} at long times and it smooths the long-time decay, thereby improving the stability of the transformed functions.

The memory kernels associated with these correlation functions are not known a priori. Although several attempts have been made [103–105], no robust, general-purpose method is currently available for extracting unknown memory kernels from noisy correlation functions, particularly when the kernels decay slowly[‡]. The framework developed here has been validated in the context of mode coupling models. Its application to direct simulation data therefore constitutes the first critical test under realistic conditions.

In this section, memory kernels are obtained directly from noisy simulation data, without explicitly enforcing theoretical assumptions about their exact functional form. This represents a major advancement, as it allows an honest and direct comparison between microscopic simulations and the predictions of mode coupling theory.

[‡]As far as I am aware.

4.4.1 Filon Method for the Inverse Laplace Transform

Before applying the neural-network-based approach, it is instructive to examine how direct inversion methods, like the Filon method for the numerical inverse Fourier–Laplace transform, see sec. 2.3.2, fail for Laplace-domain memory kernels.

The Laplace-domain representation $\widehat{S}(q,\omega)$ can be computed reliably by the forward Filon method, and the spectra appear smooth and well-behaved even for higher frequencies, as illustrated in fig. 4.25. From here, the corresponding memory kernels, $\widehat{m}(q,\omega)$, are obtained via the Laplace-domain Mori–Zwanzig equation, see eq. (4.47). However, as shown in fig. 4.26, numerical artifacts emerge at large frequencies. These artifacts are likely due to catastrophic cancellation in the Mori–Zwanzig equation, where small differences between nearly vanishing terms are amplified. Although these artifacts appear at high ω, they are particularly problematic because the inverse LT needs to integrate over the full frequency range, rendering even small high-frequency errors problematic. Truncation of the high-frequency tail of $\widehat{m}(q,\omega)$ is therefore essential to maintain stability, and this consideration directly motivated the design of the NNLT.

The limitations of the Filon method become evident when performing the inverse LT. Applying the method to $\widehat{S}(q,\omega)$ yields the results shown in fig. 4.27. While the overall shape and relaxation features of $S(q,t)$ are retained, the reconstructed function is shifted upwards. This systematic offset originates from the incompleteness of the input spectrum, where missing high-frequency contributions prevent the correct recovery of the initial condition $S(q,0) = 1$. As a consequence, even performing a forward LT directly followed by its inverse fails to preserve normalization.

The situation is even more severe for the memory kernels. The inverse transforms of $\widehat{m}(q,\omega)$, shown in fig. 4.27, perform significantly worse, particularly at short times. This failure is directly linked to the high-frequency artifacts in $\widehat{m}(q,\omega)$, which dominate the inverse LT in the small-t regime and severely distort the reconstructed kernel. At intermediate times the results improve somewhat, as the influence of the high-frequency tail diminishes.

These observations reconfirm the necessity of adopting a more robust strategy. Unlike the Filon method, the NNLT is specifically designed to deal with incomplete spectra, suppress numerical instabilities, and maintain stability even in the presence of noise. It thus provides a more flexible and accurate path for calculating memory kernels from simulation data in cases where traditional inversion methods systematically fail.

Nevertheless, the Filon method may still provide a useful complementary approach if combined with appropriate regularization or correction strategies, especially for stabilizing the short-time and high-frequency limits. A promising direction could be the integration of machine learning techniques directly into the Filon formula.

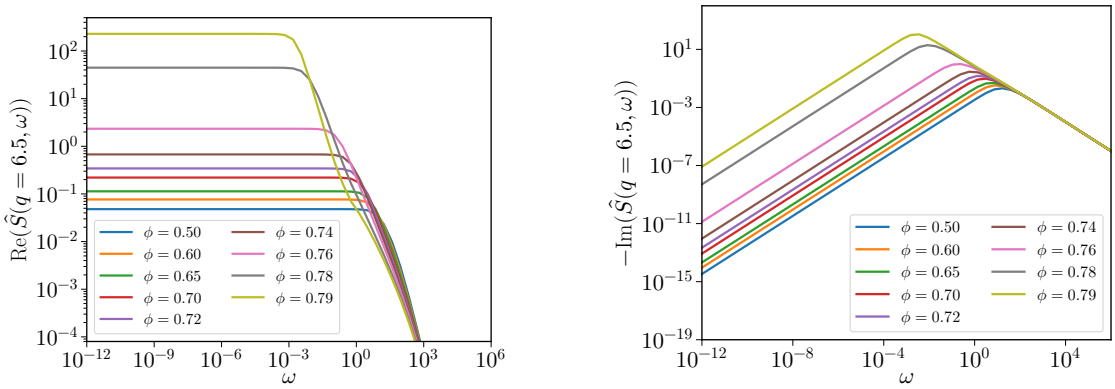


Figure 4.25: Laplace-domain self-intermediate scattering functions $\hat{S}(q,\omega)$ for Brownian particles, obtained using the Filon method for the Laplace transform for $S(q,t)$ at wave number $q = 6.5$ for different packing fractions ϕ.

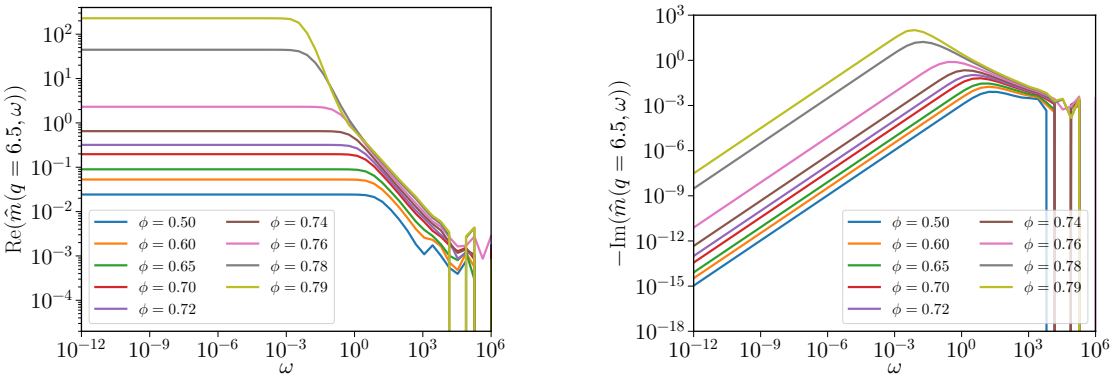


Figure 4.26: Laplace-domain memory kernels $\hat{m}(q,\omega)$ for Brownian particles, calculated from the Laplace-domain Mori–Zwanzig equation using the SISFs shown in fig. 4.25, at wave number $q = 6.5$ for different packing fractions ϕ.

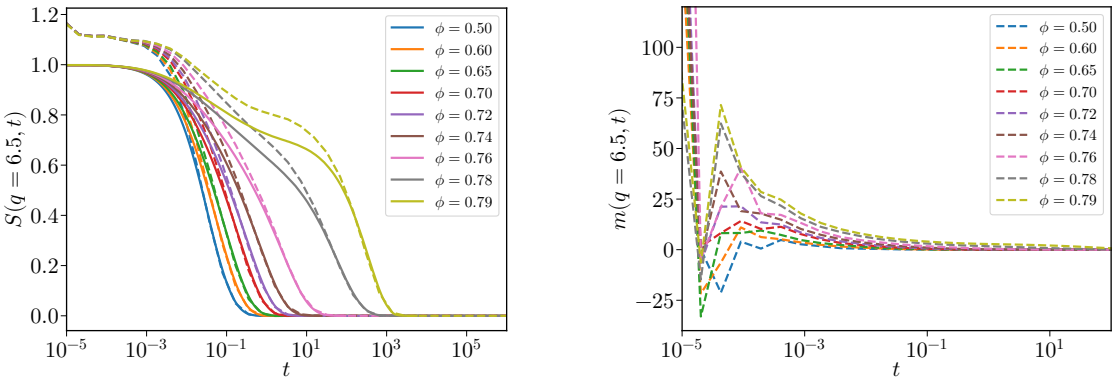


Figure 4.27: Results of the inverse Laplace transform performed with the Filon method for the SISFs (left) and memory kernels (right) at wave number $q = 6.5$ for different densities ϕ. Solid lines indicate the SISFs obtained from simulations, dashed lines the results of the Filon-transform.

4.4.2 Memory Kernel Calculation with Normalization by the Iterative Fitting Scheme

Having verified that conventional numerical inversion methods fail to accurately recover time domain memory kernels, the NNLT combined with the normalization scheme to frequency-domain data is now used for data from Brownian hard-disk simulations.

Preprocessing and Frequency Truncation

As discussed previously, the frequency-domain representation of the memory kernel $\hat{m}(q,\omega)$ obtained from simulation data exhibits numerical artifacts at high frequencies. These partially originate from catastropic cancellations in the Laplace-domain Mori–Zwanzig equation and are amplified by statistical noise in the input SISFs and its forward LT. To ensure stability, the input spectrum is truncated to $\omega < 10^2$, a cutoff that was already integrated into the architecture of the NNLT. This truncation eliminates the problematic region at the cost of removing high-frequency contributions that would be essential for estimating the initial value m_0 by asymptotic methods, see sec. 4.3. Consequently, direct normalization based on high-frequency behavior becomes unreliable[‡], and the iterative method must be employed.

Estimation of the Initial Value

The iterative normalization procedure introduced in sec. 4.3.2 is applied to determine $m_0 = m(q, t = 0)$. Candidate values for m_0 are tested, and normalized memory kernels are calculated with the NNLT. The subsequently denormalized memory kernels are then inserted into the time-domain Mori–Zwanzig equation to compute trial SISFs. By comparing the obtained correlation functions to the simulation data, the optimal m_0 can be identified through minimization of the overlap-integral error. Representative error curves for different trial values of m_0 are shown in fig. 4.28.

The qualitative features of the error curves closely resemble those observed for the Brownian hard-disk model in sec. 4.3.2. In particular, the location of minima approximately follows the expected hydrodynamic scaling $\sim 1/q^2$. In some cases the minimum splits, although this is less pronounced than for the MCT model. It should also be noted that only wave numbers for which the SISFs fully decay to zero within the simulation time are accessible, otherwise the LT does not converge.

[‡]As a guideline, the initial value estimated from the asymptotic limit is found to be approximately a factor of 3 – 4 smaller than expected, consistent with results from MCT data.

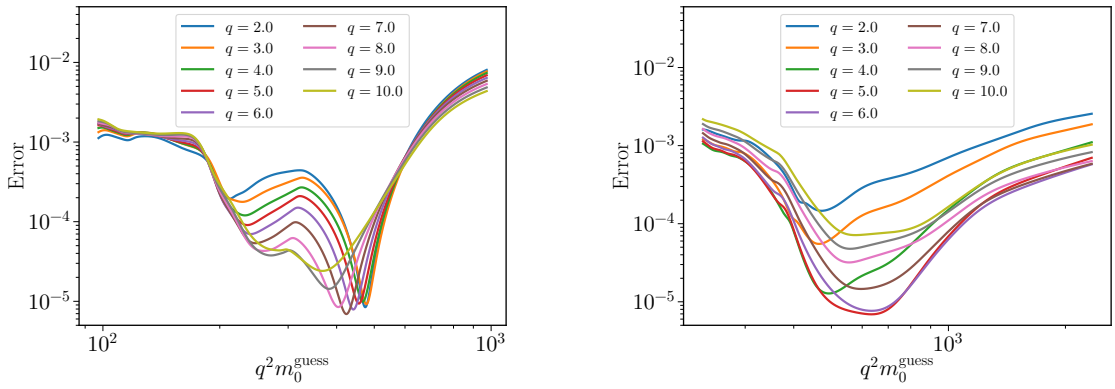


Figure 4.28: Overlap-integral error between input SISFs and SISFs calculated from the time-domain Mori–Zwanzig equation using memory kernels obtained via the Laplace-domain Mori–Zwanzig equation with the NNLT and trial normalizations for different initial value estimates m_0, shown as a function of $q^2 m_0$. Shown are packing fractions $\phi = 0.70$ (left) and $\phi = 0.77$.

Scaling Ansatz for the Initial Value

For each (ϕ, q) pair, an optimal initial value can be determined independently. However, treating each wave number in isolation introduces inconsistencies in scaling with q and, by extension, ϕ. To recover consistent scaling, the parametric scaling,

$$m_0(q) = \frac{a}{q^2}\,(1 + bq)\,, \tag{4.56}$$

with a and $b > 0$ as fit parameters, is used as introduced in sec. 4.3.2. For each packing fraction ϕ, the parameters are optimized by minimizing the average self-consistency error across all available wave numbers, i.e., wave numbers where SISFs decay to zero.

The comparison between individually optimized initial values and the best-fit scaling approach is shown in fig. 4.29 for two different packing fractions. Although the fitted scaling does not produce the exact hydrodynamic form, it enforces physically reasonable trends and stabilizes the estimation of m_0 across q. The resulting packing fraction dependence from these fits is depicted in fig. 4.30, where the scaling procedure significantly improves consistency.

For $\phi \gtrsim 0.76$, however, the scaling breaks down across all wave numbers. This density lies close to the estimated glass transition at $\phi \approx 0.78 - 0.79$, suggesting that deviations may reflect physical effects in addition to numerical limitations. Possible contributing factors include incomplete equilibration due to diverging relaxation times, the loss of short-time information from frequency truncation, and enhanced sensitivity of the normalization procedure at high densities.

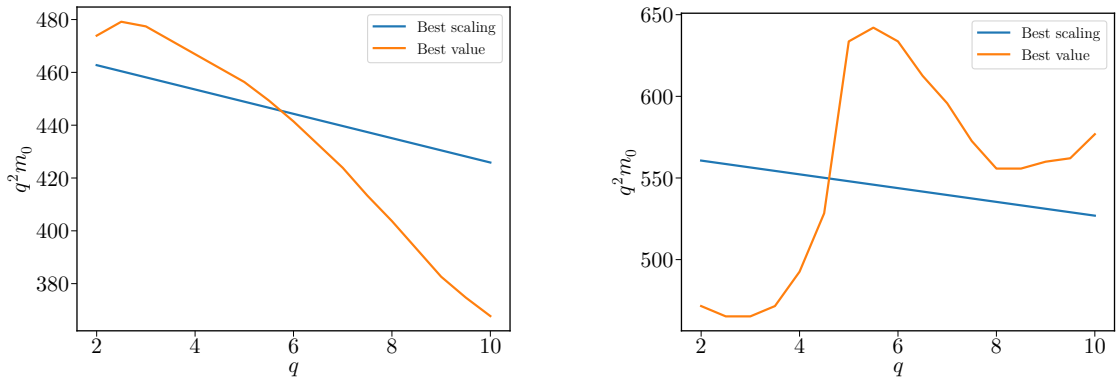


Figure 4.29: Best-fit scaling behavior of the initial value $m_0(q)$ for Brownian particles at densities $\phi = 0.70$ (left) and $\phi = 0.77$ (right), compared to individually optimized initial values obtained by minimizing the error at each wave number.

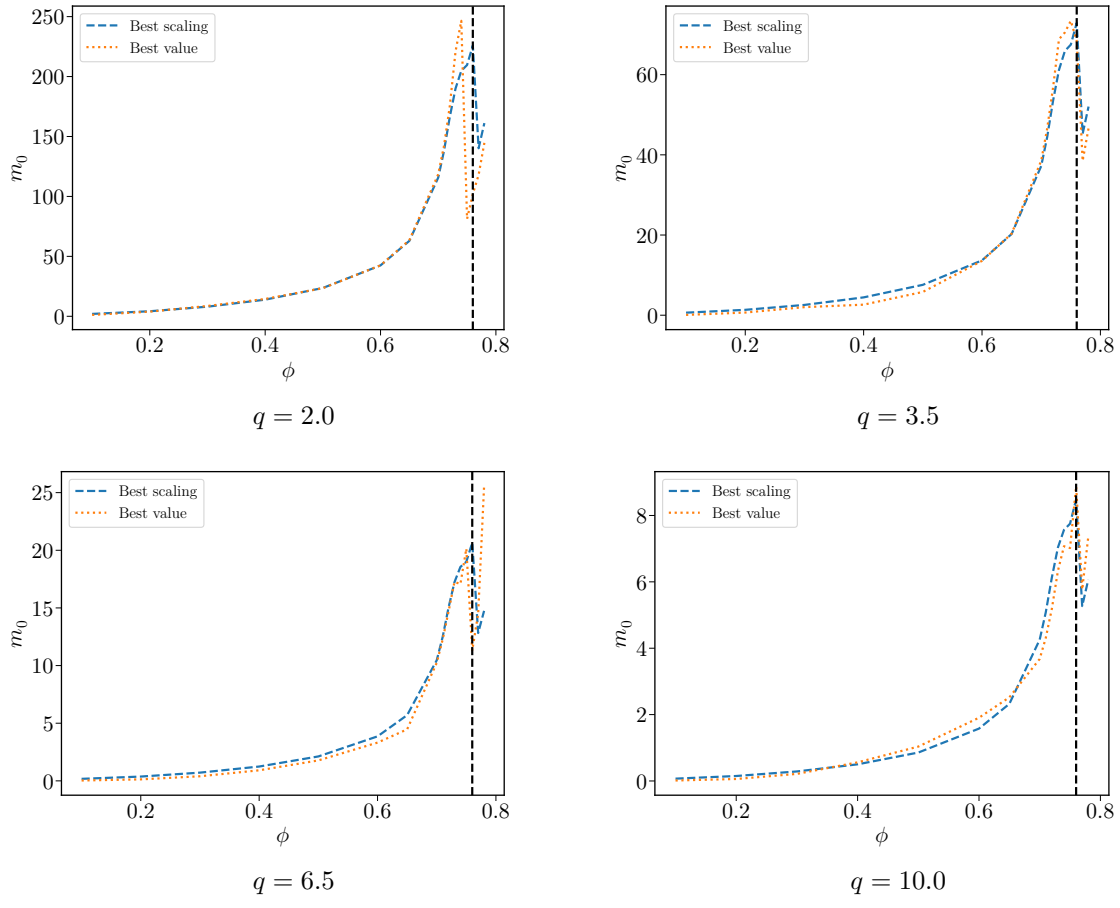


Figure 4.30: Density dependence of the initial value m_0 for Brownian particles, obtained from the iterative best-fit scaling method for $m_0(q)$, compared to individually optimized initial values at different wave numbers q. The black vertical line marks the density $\phi = 0.76$, where the scaling behavior begins to deviate from qualitative expectations.

A quantitative comparison of the self-consistency errors for the individually optimized and scaling-based normalization methods is presented in tab. 4.4. While the scaling-based method increases the error slightly, the trade-off is justified by the improved physical consistency of the results. The discrepancy in errors between the two methods is smaller than that observed for the MCT model, likely because noise, truncation, and limited resolution of the simulation data introduce a larger systematic error.

ϕ	$\mathrm{RMSE}_{\mathrm{scal}}$	$\mathrm{RMSE}_{\mathrm{best}}$
0.10	$2.506 \cdot 10^{-3}$	$2.743 \cdot 10^{-3}$
0.20	$2.156 \cdot 10^{-3}$	$1.093 \cdot 10^{-3}$
0.30	$1.341 \cdot 10^{-3}$	$7.940 \cdot 10^{-4}$
0.40	$1.579 \cdot 10^{-3}$	$9.846 \cdot 10^{-4}$
0.50	$2.045 \cdot 10^{-3}$	$1.573 \cdot 10^{-3}$
0.60	$3.107 \cdot 10^{-3}$	$2.847 \cdot 10^{-3}$
0.65	$3.444 \cdot 10^{-3}$	$3.379 \cdot 10^{-3}$
0.70	$4.177 \cdot 10^{-3}$	$3.309 \cdot 10^{-3}$

ϕ	$\mathrm{RMSE}_{\mathrm{scal}}$	$\mathrm{RMSE}_{\mathrm{best}}$
0.71	$5.441 \cdot 10^{-3}$	$3.942 \cdot 10^{-3}$
0.72	$8.449 \cdot 10^{-3}$	$6.074 \cdot 10^{-3}$
0.73	$1.394 \cdot 10^{-2}$	$9.552 \cdot 10^{-3}$
0.74	$1.236 \cdot 10^{-2}$	$9.163 \cdot 10^{-3}$
0.75	$9.536 \cdot 10^{-3}$	$8.761 \cdot 10^{-3}$
0.76	$9.709 \cdot 10^{-3}$	$8.810 \cdot 10^{-3}$
0.77	$8.026 \cdot 10^{-3}$	$6.798 \cdot 10^{-3}$
0.78	$6.322 \cdot 10^{-3}$	$5.644 \cdot 10^{-3}$

	$\mathrm{RMSE}_{\mathrm{scal}}$	$\mathrm{RMSE}_{\mathrm{best}}$
Overall	$7.053 \cdot 10^{-3}$	$5.609 \cdot 10^{-3}$

Table 4.4: RMSE between the SISFs of Brownian particles and those calculated from the time-domain Mori–Zwanzig equation using memory kernels obtained from the Laplace-domain Mori–Zwanzig equation with the NNLT and normalization by the initial value. Results are shown for initial values based on the best-fit q-scaling behavior ($\mathrm{RMSE}_{\mathrm{scaling}}$) and for individually optimized best initial values ($\mathrm{RMSE}_{\mathrm{best}}$), across different packing fractions ϕ.

Results

The resulting memory kernels calculated for both methods are shown in fig. 4.31. All memory kernels exhibit smooth shapes consistent with qualitative expectations from theory. Density scaling is markedly improved when q-scaling is enforced. In particular, at $q = 6.5$, the individually optimized method shows $\phi = 0.76$ as a clear outlier, whereas the scaling approach yields a more consistent density trend.

The corresponding SISFs obtained from solution of the time-domain Mori–Zwanzig equation are shown in fig. 4.32. Both approaches reproduce the input simulation data well, capturing the relevant relaxation time scales and dynamical features. The improvements from scaling-based normalization are thus primarily visible in the physical consistency of m_0. While minor deviations are observed in the SISFs, these cannot be solely attributed to the method of memory kernel calculation, although a multitude of possible error sources make exact identification difficult.

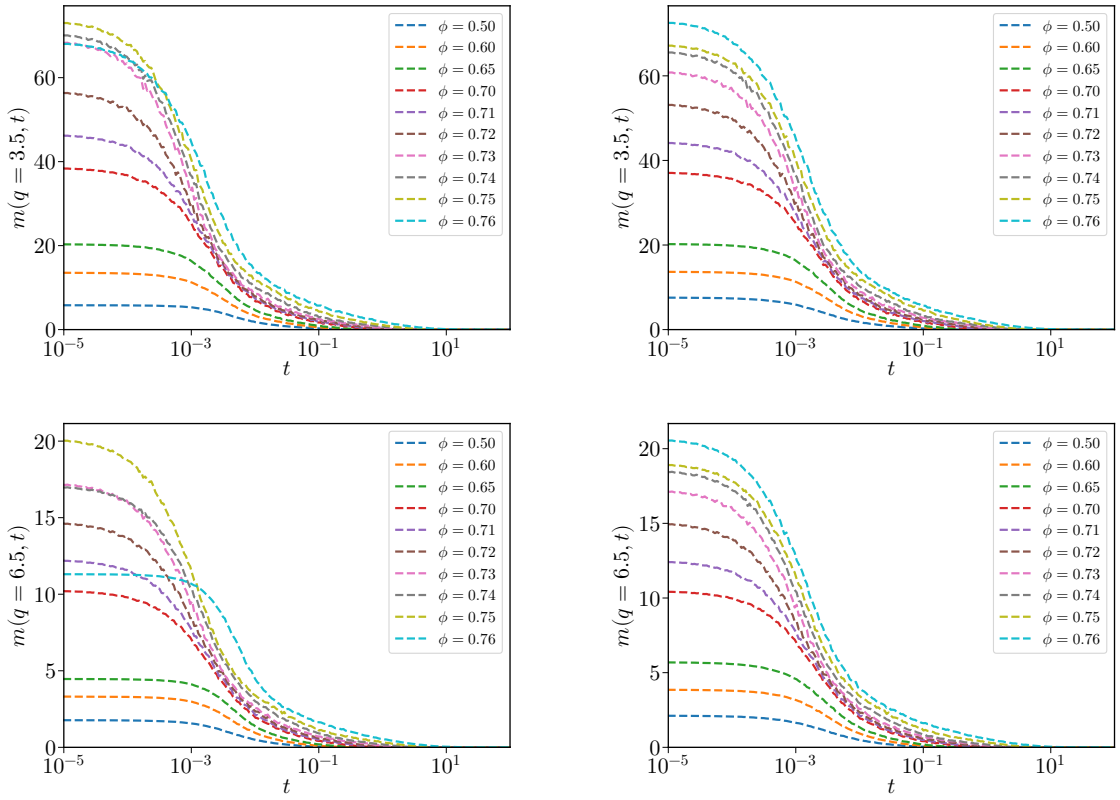


Figure 4.31: Comparison of memory kernels for Brownian particles obtained from the Laplace-domain Mori–Zwanzig equation with the NNLT using different normalization schemes for the initial value m_0. Left panels show individually optimized best values, right panels the value from best-fit q-scaling. Top panels depict wave number $q = 3.5$, bottom panels: wave number $q = 6.5$.

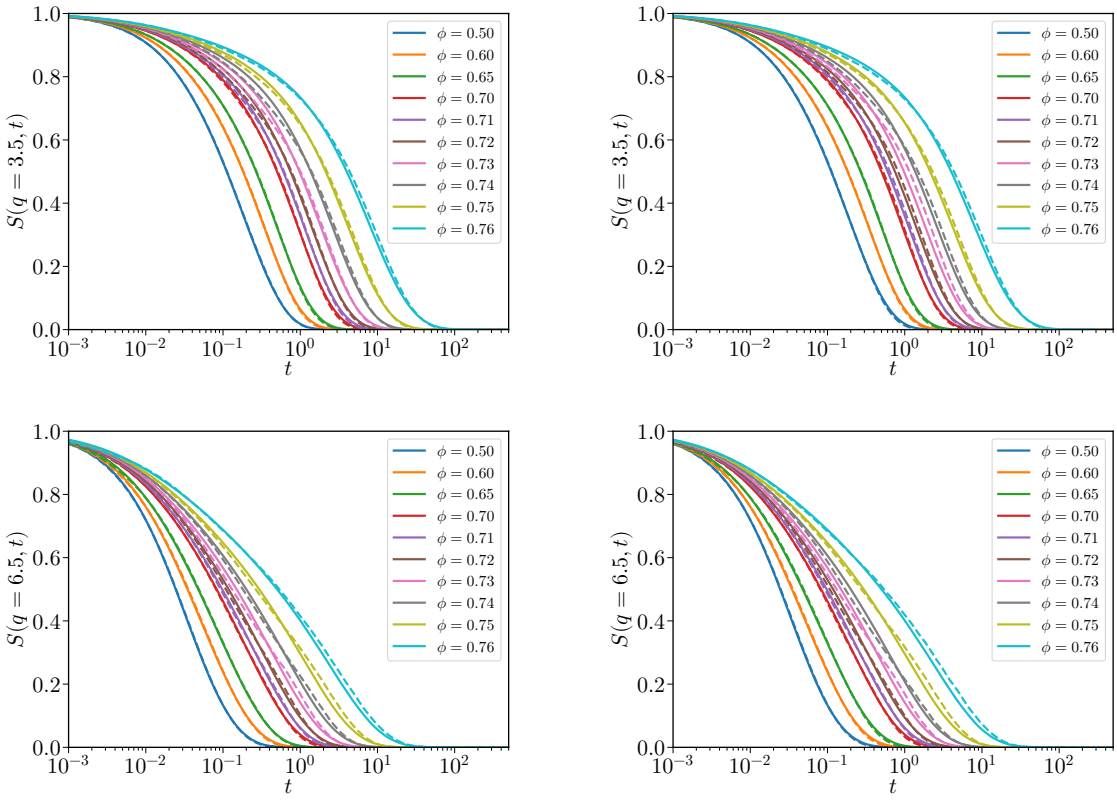


Figure 4.32: Comparison between the SISFs of Brownian particles and those obtained from the time-domain Mori–Zwanzig equation using the memory kernels shown in fig. 4.31 with different methods of estimating the initial value m_0. Left panels show individually optimized best values, right panels the value from best-fit q-scaling. Top panels depict wave number $q = 3.5$, bottom panels: wave number $q = 6.5$.

Overall, the application of the NNLT to simulation data demonstrates that reliable memory kernels can be calculated even from noisy correlation functions. The iterative normalization procedure is essential in this context, and its extension by the wave number scaling ansatz significantly improves the physical consistency of the calculated memory kernels.

Such scaling-consistent memory kernels provide the foundation for meaningful comparison with mode coupling theory and, ultimately, for constructing functionals of the memory kernel $m[S]$. The next section explores comparison with MCT in detail, providing insight into the agreement, and limitations, between theory and simulation.

4.4.3 Comparison with Mode Coupling Theory

The self-intermediate scattering functions obtained from mode coupling theory and Brownian dynamics simulations are not directly comparable. Differences in the location of the glass transition and in the scaling of packing fraction between theory and simulation prevent a one-to-one correspondence. To overcome this difficulty, SISFs predicted by MCT can be fitted to the simulation data by introducing an effective packing fraction mapping [43, 94]. This establishes a packing-fraction-conversion relation that enables comparing dynamics between theory and simulation.

Effective Packing Fraction Mapping

The resulting fits are shown in fig. 4.33, with quantitative performance summarized in tab. 4.5. Least-squared error fits were performed at $q = 6.5$, close to the first peak of the static structure factor, but other wave numbers yield basically the same scaling. At packing fractions close to the glass transition, a linear mapping suffices to align MCT and simulation results. However, at lower packing fractions, a non-linear mapping, i.e., parabolic or even higher-order in ϕ, is required, as illustrated in fig. 4.34. This reflects a fundamental difference between MCT and simulations, with the former displaying a sharp glass transition while the latter displays a more gradual slowdown of dynamics.

For packing fractions $\phi \gtrsim 0.76$, the quality of the fits deteriorates markedly. This coincides with the regime where the expected packing fraction scaling of the simulation memory kernels breaks down, see fig. 4.30. Here, MCT tends to underestimate the plateau height of the SISFs and fails to capture details of the relaxation dynamics observed in Brownian dynamics.

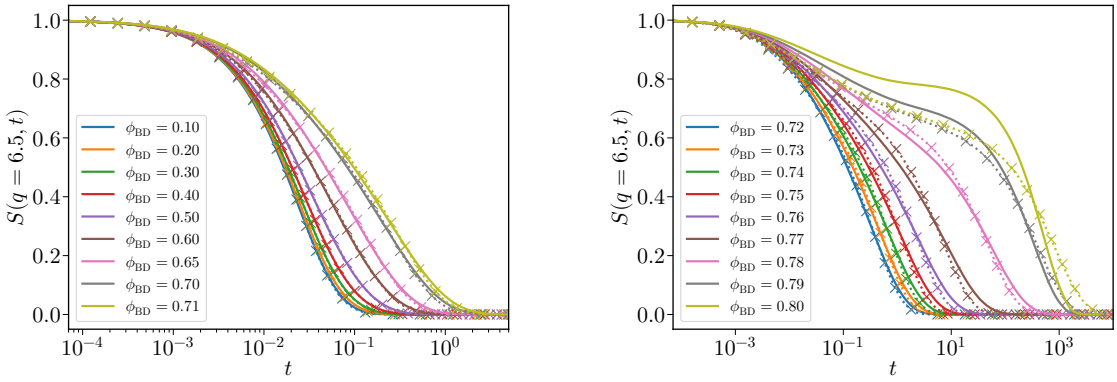


Figure 4.33: Least-squares fits of SISFs from the hard-disk MCT model (dotted with crosses) to Brownian dynamics simulation results (solid) at wave number $q = 6.5$ for multiple packing fractions ϕ_{BD}. A dense dataset of SISFs was generated within MCT to establish a reliable mapping between simulation and theoretical packing fractions.

ϕ_{BD}	ϕ_{MCT}	RMSE
0.10	0.12707	$3.593 \cdot 10^{-3}$
0.20	0.24397	$2.603 \cdot 10^{-3}$
0.30	0.33389	$1.356 \cdot 10^{-3}$
0.40	0.41225	$5.465 \cdot 10^{-4}$
0.50	0.48547	$1.971 \cdot 10^{-3}$
0.60	0.55681	$4.089 \cdot 10^{-3}$
0.65	0.59279	$5.571 \cdot 10^{-3}$
0.70	0.62951	$7.834 \cdot 10^{-3}$
0.71	0.63721	$8.619 \cdot 10^{-3}$

ϕ_{BD}	ϕ_{MCT}	RMSE
0.72	0.64473	$9.453 \cdot 10^{-3}$
0.73	0.65246	$1.064 \cdot 10^{-2}$
0.74	0.66003	$1.231 \cdot 10^{-2}$
0.75	0.66756	$1.453 \cdot 10^{-2}$
0.76	0.67487	$1.759 \cdot 10^{-2}$
0.77	0.68171	$2.151 \cdot 10^{-2}$
0.78	0.68711	$1.943 \cdot 10^{-2}$
0.79	0.68930	$2.957 \cdot 10^{-2}$
0.80	0.68966	$7.974 \cdot 10^{-2}$

Table 4.5: Resulting packing-fraction pairs $(\phi_{\mathrm{BD}}, \phi_{\mathrm{MCT}})$ obtained from the least-squares fits between SISFs of Brownian dynamics simulations and the MCT model, together with the corresponding RMSE values.

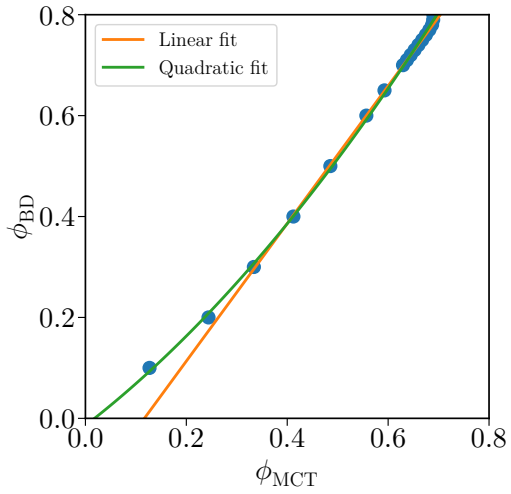


Figure 4.34: Scaling of packing fractions between Brownian dynamics ϕ_{BD} and mode coupling theory ϕ_{MCT}, determined from least-squares fits of SISFs. A linear fit captures the mapping close to dynamical arrest, while at least a parabolic fit is required to describe the full range of available packing fractions.

Comparison of Memory Kernels

With an effective mapping of packing fractions established, a direct comparison between memory kernels becomes possible. The results are presented in fig. 4.35. For MCT, memory kernels are obtained directly from the time-domain solution of the Mori–Zwanzig equation for Brownian hard-disks. For the simulation data, memory kernels are calculated using the Laplace-domain Mori–Zwanzig equation with inversion by the NNLT combined with the best-fit q-scaling normalization procedure described in sec. 4.4.2.

At effective packing fractions where SISFs are well aligned, the memory kernels agree reasonably well between theory and simulation in their intermediate- and long-time behavior. However, significant differences arise at short times, most notably in the initial values m_0. As shown in fig. 4.36, initial values m_0 obtained from the simulation memory kernels are generally smaller than the MCT predictions at low packing fractions. Around $\phi \approx 0.7$, the trend reverses and m_0 from simulations become larger than the theoretical values, marking a shift in dynamical behavior that cannot be accounted for by simple mapping.

At very high packing fractions, the discrepancies become more pronounced, and the fits lose reliability. Outliers at $\phi = 0.77$ and $\phi = 0.78$ are excluded from detailed analysis, as they cannot be meaningfully compared. Packing fractions above this point are omitted entirely, since it is well established that MCT fails to reproduce dynamics close to the glass transition in both simulations [106] and experiments [107].

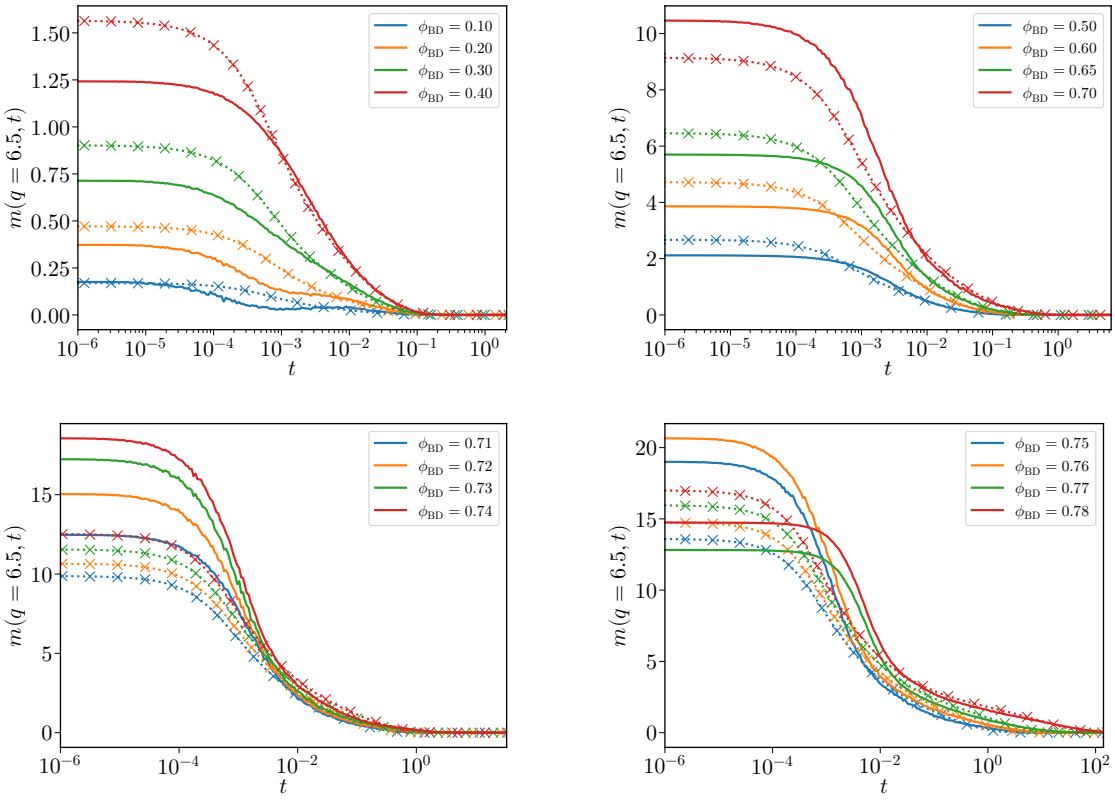


Figure 4.35: Comparison of memory kernels from mode coupling theory for Brownian hard-disks with those calculated for Brownian dynamics simulations at wave number $q = 6.5$ for multiple packing fractions ϕ_{BD}. The corresponding mode coupling packing fractions are given in tab. 4.5. Solid lines show results from simulations obtained with NNLT, while dotted lines with crosses represent the theoretical memory kernels.

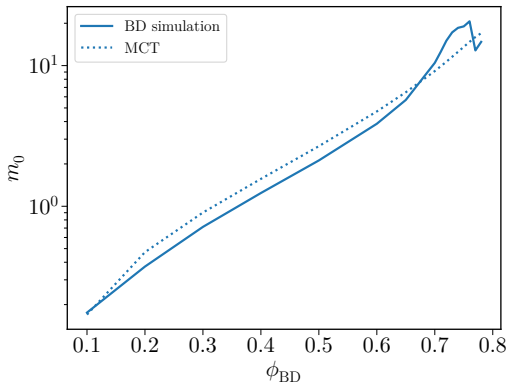


Figure 4.36: Initial values m_0 of the memory kernel as a function of Brownian dynamics packing fraction ϕ_{BD}, compared to mode coupling theory results. Solid lines indicate simulation data, while dotted lines show the MCT predictions plotted against the scaled packing fraction.

Discussion

The comparison highlights some strengths and limitations of mode coupling theory for Brownian particles. On the one hand, MCT reproduces features of the memory kernel at intermediate times, consistent with its success in describing aspects of glassy dynamics. On the other hand, it misses essential contributions to the dynamics, as reflected in the discrepancies of the short-time initial values m_0 and in its failure at high packing fractions closest to dynamical arrest.

It should also be noted that the m_0 values obtained for simulation data remain somewhat ambiguous due to numerical limitations. Attempts to use the MCT initial values directly in the inversion procedure did not improve agreement. Instead, they worsened the comparison, especially in the intermediate-time regime. This proves that the observed behavior is not solely due to normalization and stems from fundamental differences between the dynamics of MCT and simulation.

In summary, memory kernels from simulations and theory show partial qualitative agreement but diverge in some aspects. In particular, prediction of an ideal glass transition of MCT is not observed in simulations. Improving the quality of simulation data, for example by extending trajectory lengths or enhancing ensemble averaging, would help to clarify which differences originate from numerical noise and which reflect fundamental physical phenomena beyond the scope of MCT.

Summary

In this section, the method for calculating memory kernels from correlation functions was applied to Brownian dynamics simulations. Despite statistical noise and finite length of simulation trajectories, the NNLT proved robust in performing the inverse LT of the memory kernel obtained from the Laplace-domain Mori–Zwanzig equation, yielding reliable time-domain memory kernels.

The calculated memory kernels allowed for a direct comparison with predictions from mode coupling theory. After correcting for the known mismatch in packing fraction scaling between simulations and MCT, good agreement was observed for small and moderate packing fractions, particularly in the intermediate- and long-time regime. Nevertheless, systematic discrepancies appeared at short times. These can be traced to a possible underestimation of initial values m_0, the restricted frequency windows available for the LT of simulation data, and contributions absent in the theoretical description.

Closer to the glass transition, deviations between simulation and theory became more pronounced. The simulation memory kernels exhibit qualitatively different features from

their MCT counterparts, in line with the shortcomings already visible in the comparison of self-intermediate scattering functions. While MCT does capture slow dynamics, it fails to fully reproduce the microscopic mechanisms of dynamical arrest observed in Brownian dynamics simulations.

Having obtained memory kernels over a broad range of densities and wave numbers for Brownian particles, these can now be used in an attempt to construct a mode-coupling-like closure, $m[S]$, directly from simulation data. This approach opens the possibility of deriving closures that go beyond existing theory and provide new insights into the physics of glassy dynamics.

4.5 Neural Network Functional for Memory Kernels

The results of the previous sections have demonstrated that memory kernels can be reliably calculated for Brownian dynamics simulations using the Laplace-domain Mori–Zwanzig equation in combination with the NNLT and the iterative normalization scheme. This enables constructing a relation $m_\phi[S]$ that expresses the memory kernel as a functional of the correlation functions, with explicit dependence on the packing fraction ϕ. In standard MCT, such a closure is obtained via the mode coupling approximation, yielding a quadratic functional of the correlation functions. While this property is only stricly true for the ISF in MCT, the SISF is used here as a proof-of-concept, since the ISF and SISF exhibit qualitatively similar behavior.

For simulation data, however, the existence or form of such a functional relation are not known. With accurate memory kernels from simulations now available, it becomes possible to probe whether an effective closure can be learned directly from the tuples $(S(q,t), m(q,t), \phi)$ obtained in the previous section.

To approximate this functional, a simple feed-forward neural network is employed, making use of its capability as a universal function approximator. As a first step, the method will be validated on data generated from MCT, where the functional is explicitly known and the network accuracy can be tested systematically. Once validated, the approach will be applied to Brownian dynamics simulations to infer an empirical closure. Deviations from the MCT predictions may then be interpreted as signs of additional physical mechanisms absent from the theory, potentially offering new insights into the microscopic origin of glassy dynamics.

4.5.1 Network Architecture

To learn the relation between SISFs and memory kernels, a feed-forward neural network is employed. Since the functional depends explicitly on the packing fraction ϕ, it is included as part of the network input. The training data consist of tuples $(S(q), m(q), \phi)_t$, with each time point t constituting an independent sample.

Both $S_t(q)$ and $m_t(q)$ are discretized on a fixed q-grid with N_q points. Accordingly, the input to the network is a vector of the form

$$\{S_t(q_1), \ldots, S_t(q_{N_q}), \phi\}, \tag{4.57}$$

while the output corresponds to the discretized memory kernel,

$$\{m_t(q_1), \ldots, m_t(q_{N_q})\}. \tag{4.58}$$

In the present implementation, the q-grid contains $N_q = 9$ values ranging from $q = 2.0$ to $q = 10.0$, chosen to capture the most accessible regime for Brownian dynamics simulations.

The neural network architecture is fully connected, consisting of three hidden layers with 32, 16, and 32 neurons, respectively. Hyperbolic tangent (tanh) activation functions are employed in the hidden layers to provide non-linearity. The output layer uses a modified exponential linear unit (ELU) activation, which is exponential for small values near zero and linear for values above one. This choice is motivated by the physical properties of the memory kernel, which is non-negative yet unnormalized.

Training is carried out using mini-batch gradient descent with the ADAMW optimizer, combined with an exponentially decaying learning rate schedule. Learning rate decay is triggered when the validation loss saturates, ensuring convergence to a stable solution. Each available time point t in the dataset is treated as an independent training sample, reflecting the time-independent nature of the closure $m_\phi[S]$. The training data spans a broad range of packing fractions, thereby enabling the network to learn both the structural and packing fraction dependence of the functional.

Once trained, the neural network represents an effective functional relation for the memory kernel. It can be inserted into the time-domain Mori–Zwanzig equation to iteratively solve for the SISF. This framework provides a flexible and general approach to predicting dynamical behavior, inspired by the mode coupling approximation, but allowing the possibility of uncovering functional forms for $m_\phi[S]$ that go beyond the assumptions of standard MCT.

4.5.2 Neural Network Functional for Mode Coupling Theory

For Brownian hard-disk systems, MCT provides an explicit analytical relation for the memory kernel $m_\phi[S]$, based on the static structure factor derived from FMT, see secs. 4.1.4 and 4.1.5. This makes the model an ideal benchmark for testing the neural network functional, since both the SISF and the memory kernels can be computed to high numerical precision across a wide range of wave numbers q and packing fractions ϕ.

Given the availability of clean and consistent data, training the neural network is straightforward. The problem is well-posed, and the mapping between inputs, i.e., discretized SISFs and packing fraction ϕ, and outputs, i.e., discretized memory kernels, is numerically

robust. Variations in performance are therefore expected to be primarily due to stochastic fluctuations of the optimization process rather than limitations of the architecture itself. The trained network reproduces the mode coupling functional with low error and can be further validated by embedding it into the Mori–Zwanzig solver mctpy [92], replacing the mode coupling approximation functional.

The non-ergodicity parameter is a key indicator of the onset of glassy dynamics in a system [97, 98]. It is defined as the long-time limit of the self-intermediate scattering function,

$$f(q) = \lim_{t\to\infty} S(q,t). \tag{4.59}$$

A non-vanishing value, i.e., $f(q) > 0$, signals dynamical arrest. The non-ergodicity parameter satisfies the algebraic equation

$$m[f](q) = \frac{f(q)}{1-f(q)}. \tag{4.60}$$

The calculation of $f(q)$ therefore constitutes a central benchmark for evaluating the quality of any proposed functional relation $m_\phi[S]$, and $f(q)$ can be obtained numerically using the mctpy package [92], allowing a direct comparison between analytical mode coupling predictions and those generated by the neural network functional.

Fig. 4.37 compares the non-ergodicity parameters obtained from MCT to those obtained with the neural network approximation for the memory kernel. Both methods yield essentially identical predictions for the critical point, ϕ_c, confirming that the network is able to capture the glass transition. Deviations in the value of $f(q)$ are observed for increasing wave numbers, reflecting the sensitivity of the non-ergodicity parameter to small variations near the bifurcation. The network does not encode this feature explicitly, and is only able to learn it implicitly from the training data. Incorporating packing fraction reparameterizations or regularization strategies could improve the robustness of the network functional close to the critical point.

Beyond the non-ergodicity paramater, the neural network functional can be used to produce full time-dependent correlation functions and memory kernels. Figs. 4.38 and 4.39 present comparisons of SISFs and memory kernels obtained by solving the time-domain Mori–Zwanzig equation with either the mode coupling approximation or the neural network functional. The first figure varies packing fraction at fixed q, while the second varies q at fixed packing fraction. Although the memory kernels produced by the two closures differ only slightly, these differences lead to visible changes in the SISF, especially near the glass transition. This highlights the well-known sensitivity of the Mori–Zwanzig equation to fine details of the memory kernel.

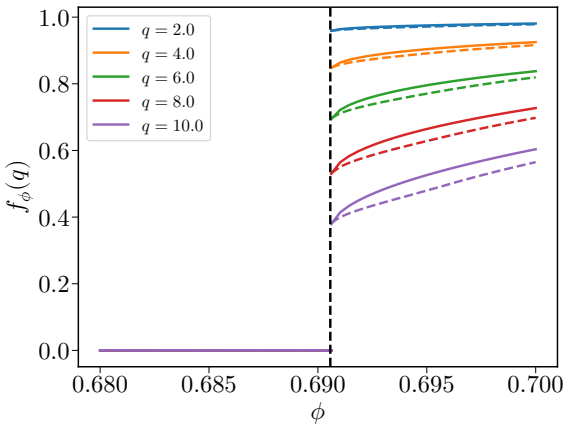


Figure 4.37: Non-ergodicity parameter $f_\phi(q)$ as a function of packing fraction ϕ for different wave numbers q. Solid lines show results obtained with the mode coupling approximation of Brownian hard-disks, while dashed lines indicate predictions from the neural network functional. The vertical dashed line marks the critical packing fraction $\phi_c = 0.69057$ of the glass transition.

The performance of the neural network closure is particularly limited in two regimes. First, near the critical point, small deviations in the input cause large discrepancies in the predicted dynamics. Second, for very small kernel values, i.e., $m(q,t)/m_0(q) \lesssim 10^{-3}$, accuracy decreases due to the interplay of limited numerical precision and the unbounded range of the output space. These shortcomings may be alleviated by employing more expressive architectures or by reparameterizing the memory kernel to reduce sensitivity in these regimes.

Despite these limitations, the neural network functional shows strong overall performance. It successfully reproduces important features of the mode coupling approximation, including the location of the glass transition, the form of the non-ergodicity parameter, and the qualitative structure of the dynamics across multiple packing fractions and wave numbers. Neural network functionals thus provide an accurate surrogate for the mode coupling approximation. Nevertheless, the sensitivity to critical behavior and numerical instabilities necessitates careful treatment. Importantly, the successful reproduction of the mode coupling approximation validates the approach and establishes its feasibility for application to simulation data, where no such functional is known.

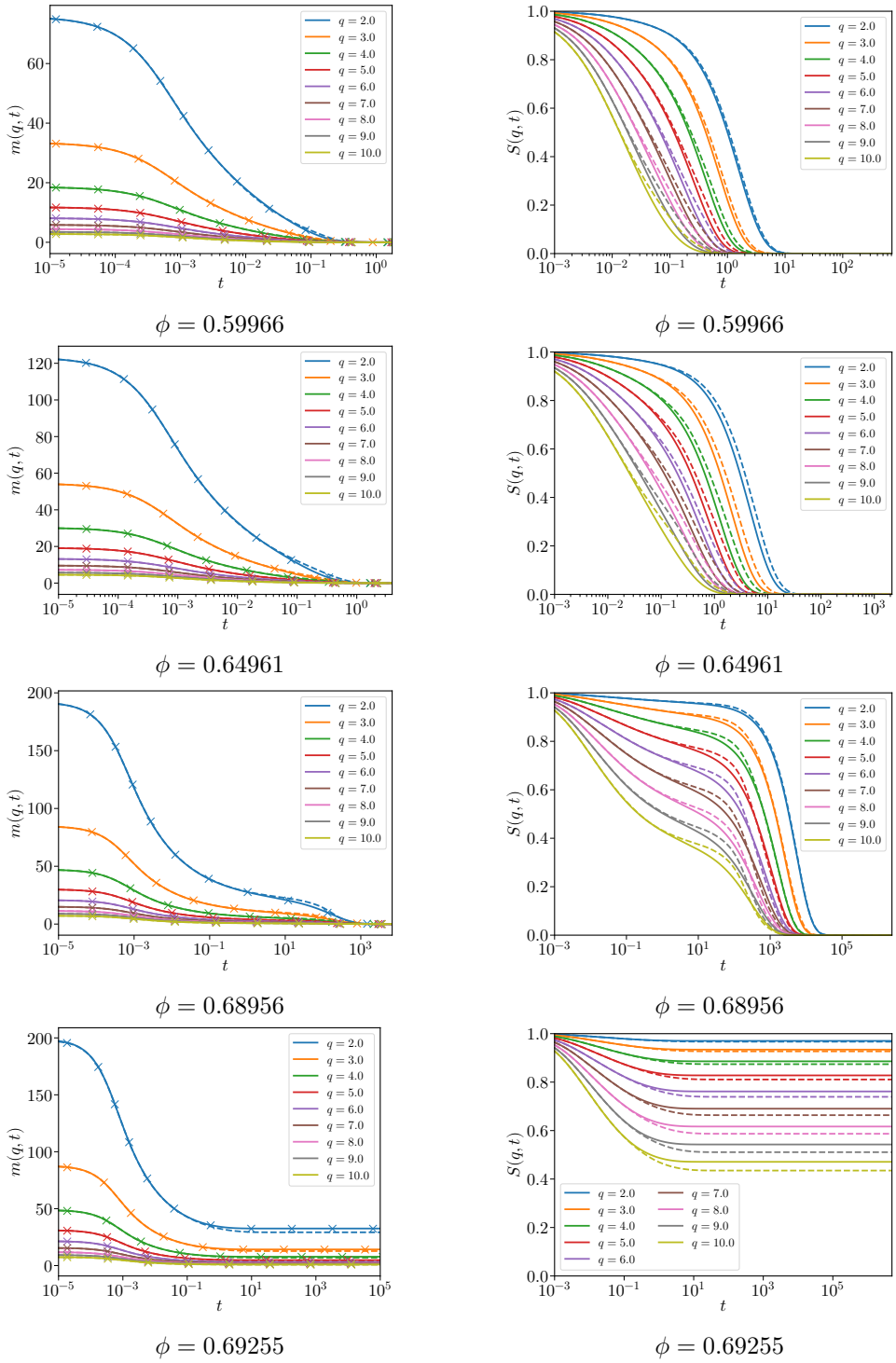


Figure 4.38: Solutions of the time-domain Mori–Zwanzig equation using the mode coupling approximation for Brownian hard-disks (solid) and the neural network functional (dashed) for different wave numbers q at fixed packing fraction ϕ. Crosses indicate the memory kernels obtained directly from the network when supplied with the SISFs from mode coupling theory as input. The q-values correspond to the discretized q-grid used for the network input and output.

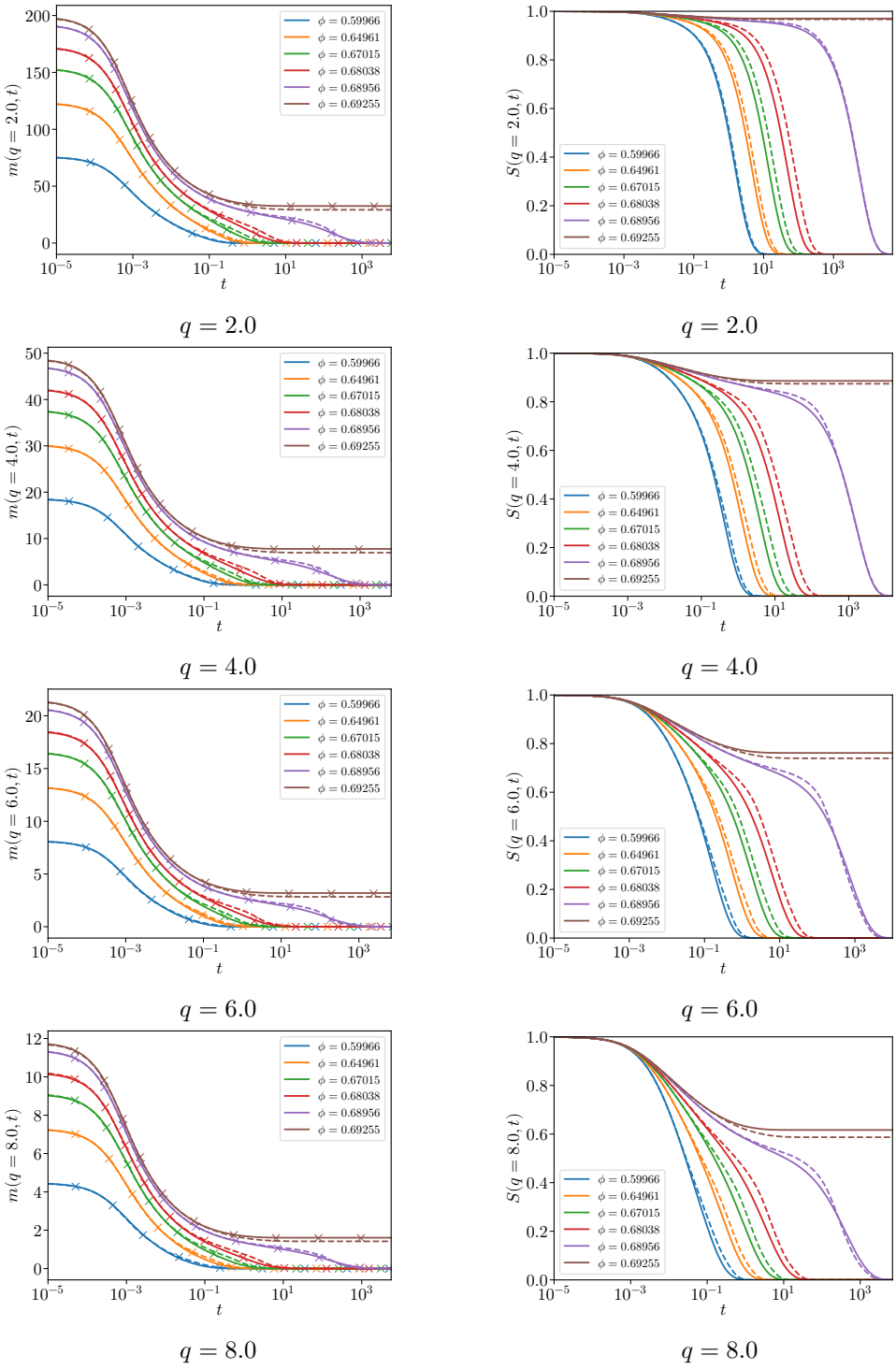


Figure 4.39: Solutions of the time-domain Mori–Zwanzig equation using the mode coupling approximation for Brownian hard-disks (solid) and the neural network functional (dashed) for different packing fractions ϕ at fixed wave numbers q. Crosses indicate the memory kernels obtained directly from the network when supplied with the SISFs from mode coupling theory as input. The q-values correspond to the discretized q-grid used for the network input and output.

4.5.3 Neural Network Functional for Brownian Particles

Whether a well-defined functional $m_\phi[S]$ exists for simulation data, analogous to the mode coupling approximation, remains an open question. Nevertheless, the neural network framework introduced in the previous section provides a systematic means of probing this possibility by attempting to learn a mode coupling-like closure directly from simulation data.

Applying this approach to Brownian dynamics data introduces several challenges absent in the theoretical case. Most critically, the available dataset is much more limited, since only a limited amount of packing fractions are accessible due to the long simulation times required. Increasing temporal resolution alone does not necessarily improve richness of training data. Furthermore, the data is inherently noisy and relies on the successful calculation of memory kernels.

To mitigate these issues, it is essential to ensure that the training data exhibit physically reasonable q- and ϕ-scaling behavior. This is achieved by enforcing the normalization of the memory kernels through the iterative scheme, stabilizing the initial value m_0 across most packing fractions. Nevertheless, data at high packing fractions, i.e., $\phi > 0.76$, had to be excluded, as deviations from the expected density scaling resulted in significantly degraded network performance.

The training data consist of tuples $\{S_t(q_1), \ldots, \phi\}$ as inputs and the corresponding memory kernels $\{m_t(q_1), \ldots\}$ as targets, both discretized on the same q-grid used for the MCT-based network. The same architecture, loss function, and optimization are employed, with each available time point t treated as an independent training sample.

It is important to note that the memory kernels obtained via the NNLT do not reproduce the original simulation SISFs exactly. Consequently, the network does not learn a perfectly consistent functional, but rather an approximate relation subject to statistical noise. This approximation is deemed acceptable, provided the network can capture the essential structure of the data.

As in the case of the mode coupling approximation and the MCT neural network functional, the non-ergodicity parameter $f(q)$ can be computed from the resulting simulation functional. Results are shown in fig. 4.40. Although no analytical reference is available, the results can be evaluated against the approximate onset of dynamical arrest in simulations, which occurs somewhere between $\phi = 0.78$ and $\phi = 0.79$, depending on the applied criterion. Notably, the network was trained only up to $\phi = 0.76$, so predictions at higher densities represent extrapolation. Despite this, the learned functional predicts the transition region reasonably well, though with a smoother crossover than MCT, possibly consistent with the absence of an ideal glass transition.

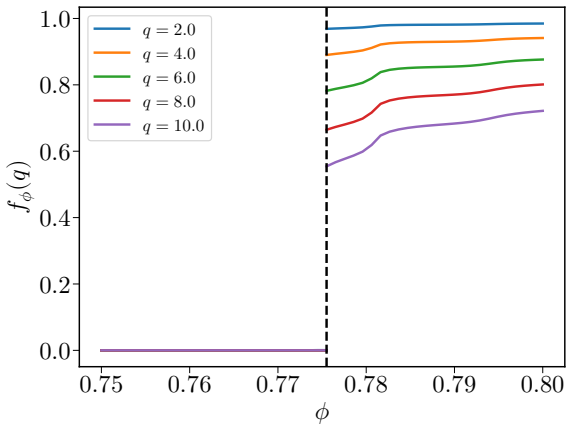


Figure 4.40: Non-ergodicity parameter $f_\phi(q)$ obtained from the neural network functional as a function of packing fraction ϕ for different wave numbers q. The expected onset of dynamical arrest occurs above $\phi \approx 0.78$, while predictions for $\phi > 0.76$ represent extrapolation beyond the training range. The vertical black line marks the density $\phi = 0.776$ at which the non-ergodicity parameter exhibits a jump.

The predictive power of the neural network functional is further tested by solving the time-domain Mori–Zwanzig equation for SISF and memory kernel. Figs. 4.41 and 4.42 compare the resulting SISFs to simulation results and memory kernels to those obtained by the Laplace-domain Mori–Zwanzig equation with NNLT, varying q at fixed packing fraction ϕ and varying ϕ at fixed wave number q, respectively.

The network demonstrates excellent agreement within its training range. Both the SISFs and the memory kernels reproduce the relevant time scales and relaxation features well. Even at $\phi = 0.77$, the first packing fraction beyond those provided in the training set, the predictions remain qualitatively accurate to some degree. This indicates that the neural network has learned some generalization of the functional and did not just perform a pointwise curve fit.

Nevertheless, some limitations remain. Simply inserting the memory kernels obtained by NNLT into the Mori–Zwanzig equation already produces moderate deviations from the input SISFs, as discussed in sec. 4.4.2, and these propagate into the learned functional. Moreover, variability between training runs showed some sensitivity of the learned functional to stochastic training effects.

Overall, these results demonstrate that the neural network functional can interpolate reliably within its training range and also extrapolates meaningfully, capturing important dynamical features of the system. The ability to learn such a functional from simulation data provides strong evidence that a functional relation $m_\phi[S]$ inspired by the mode coupling approximation exists for Brownian particles, reinforcing the connection between mode coupling theory and simulations.

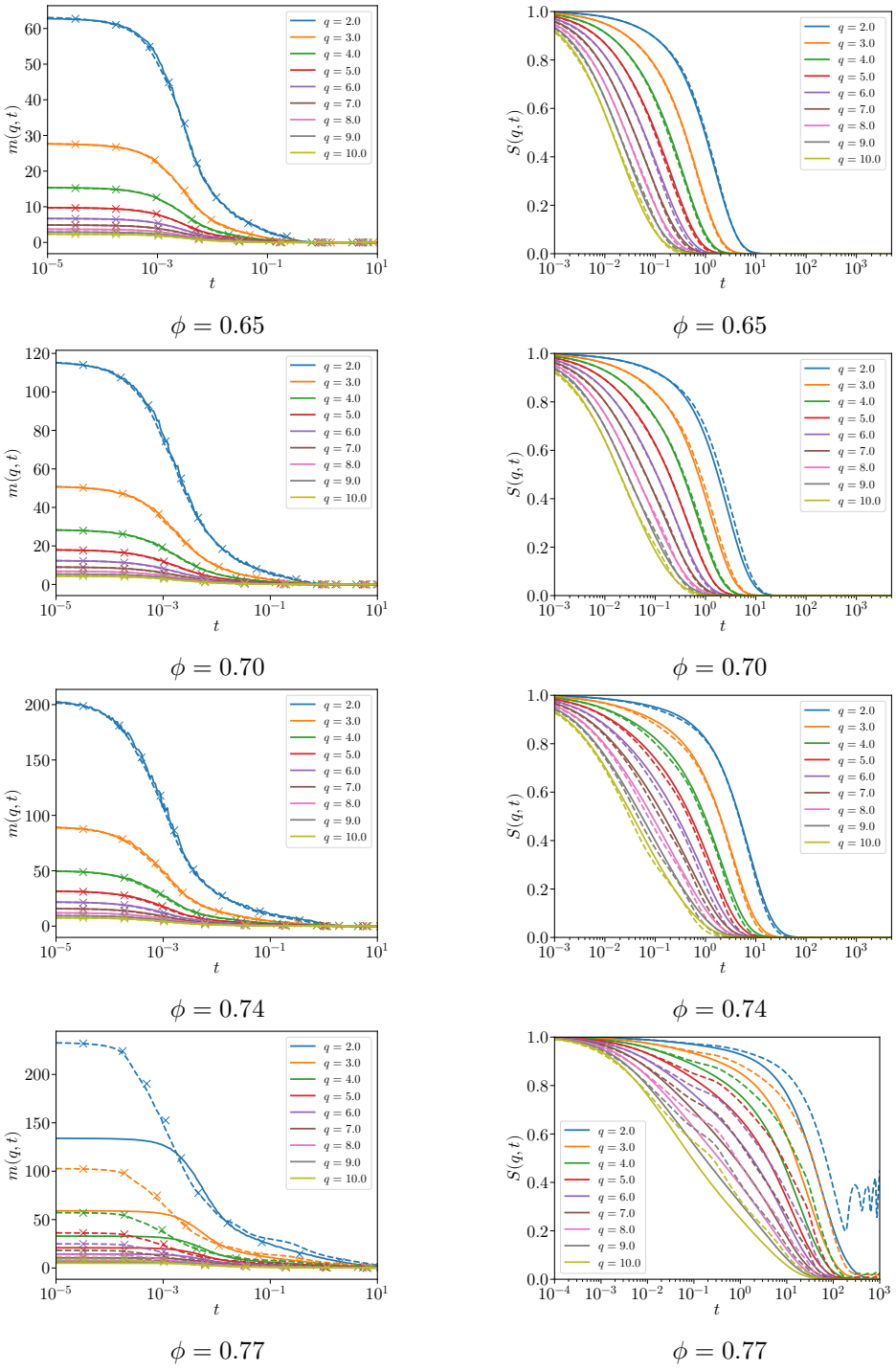


Figure 4.41: Self-intermediate scattering functions (right) and memory kernels (left) for Brownian particles (solid) compared to solutions of the time-domain Mori–Zwanzig equation using the neural network functional (dashed) at fixed packing fractions ϕ and different wave numbers q. The memory kernels were obtained from the Laplace space Mori–Zwanzig equation combined with the NNLT and normalization scheme. Crosses indicate memory kernels predicted directly by the functional when provided with SISFs from mode coupling theory as input. The q-values correspond to the discretized grid used for the network.

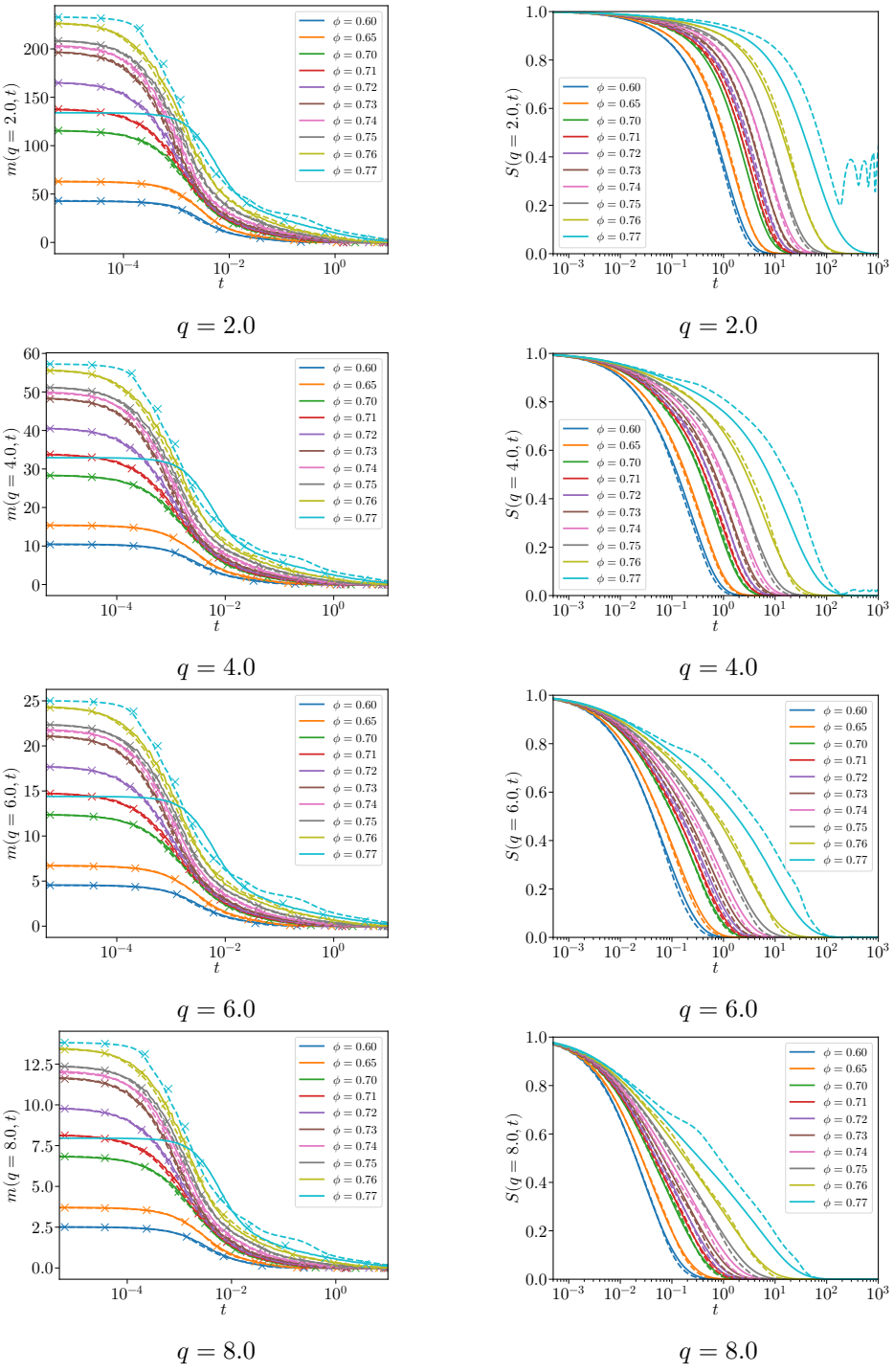


Figure 4.42: Self-intermediate scattering functions (right) and memory kernels (left) for Brownian particles(solid) compared to solutions of the time-domain Mori–Zwanzig equation using the neural network functional (dashed) at fixed wave numbers q and different packing fractions ϕ. The memory kernels were obtained from the Laplace space Mori–Zwanzig equation combined with the NNLT and normalization scheme. Crosses indicate memory kernels predicted directly by the functional when provided with SISFs from mode coupling theory as input. The q-values correspond to the discretized grid used for the network.

4.5.4 Neural Network Memory Kernel Closure Comparison between Theory and Simulation

Neural network functionals for the memory kernel have been successfully constructed for both mode coupling theory and Brownian dynamics simulations. Since both networks share the same architecture, were trained using identical procedures, and operate on the same discretized grid, their outputs are directly comparable once the packing fraction scaling between the two systems is properly accounted for. The necessary mapping between simulation packing fraction ϕ_{BD} and the mode coupling packing fraction ϕ_{MCT} is provided by the fitting procedure described in sec. 4.4.3.

To obtain a neutral benchmark for comparison, both functionals are evaluated on a smooth test input,

$$S(q) = \frac{a}{1 + bq^2}, \tag{4.61}$$

with parameters a and b chosen to produce a monotonically descreasing structure factor. This dummy input avoids fluctuations or noise present in real data. The outputs of the two networks can then be compared directly, as shown in fig. 4.43.

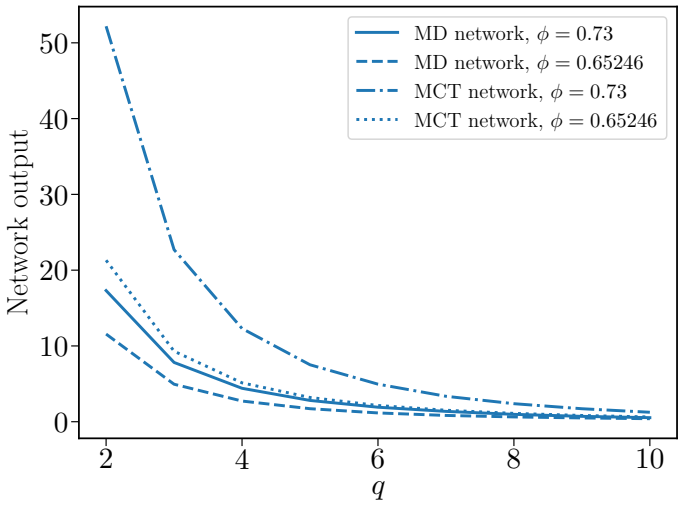


Figure 4.43: Output of the neural network functional for mode coupling theory and Brownian dynamics simulations using the test input $S(q) = \frac{1}{1+0.01q^2}$. The results are shown for the mapped density pair $(\phi_{\mathrm{BD}} = 0.73, \phi_{\mathrm{MCT}} = 0.65246)$.

The most prominent difference is observed in the magnitude of the predicted memory kernels. The simulation-trained network consistently produces smaller values than its mode-coupling counterpart, with the deviation approximately resembling a constant scaling factor that diminishes slightly with increasing wave number q. By contrast, the overall

shape and q-dependence of the predicted memory kernels remain similar across the two closures.

Because both networks have already been validated against their respective ground truths, i.e., mode coupling approximation for MCT and NNLT-calculated memory kernels for simulations, these discrepancies are attributed to physical differences between theory and simulation. These include the absence of a sharp ideal glass transition and differing packing fraction dependence close to dynamical arrest.

This analysis confirms that the network functionals have indeed learned the characteristic structure of the relations within their respective domains. At the same time, the comparison highlights that the trained networks also encode the deeper, systematic differences between MCT and real-world particle systems.

Summary

In this section, a neural network functional for the memory kernel was developed. The approach was first benchmarked on the mode coupling approximation, where the functional $m_\phi[S]$ is known analytically. The network reproduced the relation reasonably well, being able to both reproduce the non-ergodicity parameter and the critical density associated with dynamical arrest. This served as a proof of concept, demonstrating that a feed-forward neural network can successfully approximate the mode coupling functional and reproduce its dynamical predictions when solving the time-domain Mori–Zwanzig equation.

The methodology was then used for data of Brownian dynamics simulations, where the construction of such a closure had not previously been possible due to the absence of reliable memory kernels. With the NNLT, memory kernels could be calculated from the self-intermediate scattering functions via the Laplace-domain Mori–Zwanzig equation, enabling direct training of a neural network functional on simulation data. The resulting network showed convincing interpolation within the training range and even produced some meaningful extrapolations, correctly predicting the smooth onset of dynamical arrest observed in simulations, albeit at a slightly lower packing fraction.

A comparison between the functionals obtained from theory and simulation revealed qualitative similarities in their structure. The primary differences show in physical behavior of the two systems. In particular, the simulation-based closure predicted systematically smaller memory kernel amplitudes and exhibited a smoother density dependence. These differences highlight the capacity of functional derived from simulation data to capture features beyond the reach of standard theoretical approximations.

Overall, this section establishes that neural networks provide a powerful tool for constructing functionals for memory kernels. They reproduce established theoretical results and extend

well to simulation data. This opens the way towards developing hybrid approaches, where insights from theory and simulation can be combined to build more accurate and general descriptions of glassy dynamics.

4.6 Summary and Outlook

This chapter introduced the method for the neural network inverse Laplace transform (NNLT) of correlation functions, with a particular focus on applications to Brownian dynamics. Trained on synthetic data composed of superpositions of decaying exponentials, the NNLT demonstrated strong generalization capability, accurately reconstructing time-domain correlation functions from their Laplace-domain representations from both mode coupling theory and simulation data.

Building on this foundation, the NNLT was embedded into a framework for calculating memory kernels from correlation functions. The correlation function is first transformed into Laplace space, where the memory kernel is obtained using the Laplace-domain Mori–Zwanzig equation. The corresponding time-domain representation is then recovered via the NNLT, yielding reliable time-domain memory kernels. To address the unnormalized nature of memory kernels and stabilize the inversion, an iterative normalization scheme was introduced. This procedure was refined by imposing hydrodynamic scaling laws, ensuring physically consistent normalization.

This method enabled the calculation of memory kernels directly from self-intermediate scattering functions of Brownian dynamics simulations (see fig. 4.31), allowing direct comparison with predictions of mode coupling theory. This revealed systematic deviations in short-time structure and density scaling, reinforcing the well-known limitations of MCT in capturing the dynamics near arrest.

The calculated memory kernels then provided the basis for constructing mode-coupling-like neural network functionals, $m_\phi[S]$. First validated on MCT, where this relation is given by the mode coupling approximation, the method was then extended to simulation data, demonstrating that such a functional mapping exists in practice. The resulting functional could reproduce essential dynamical features, thereby offering new ways to investigate the physics of glassy dynamics.

Looking forward, neural network functionals derived from simulations offer a promising route towards more accurate models of glass-forming systems. They provide a systematic framework for comparing theory and simulation and have the potential to guide theoretical developments where existing approximations break down. The methodology established in this chapter therefore opens the way towards hybrid approaches that combine the predictive structure of MCT with the accuracy of simulations.

The next step is the prediction of memory kernels for the intermediate scattering function. The extension from the self-intermediate scattering function to the ISF is expected to be straightforward. Within MCT, memory kernels of ISFs were already extracted with

accuracy comparable to that obtained for the SISF, and the same behavior is expected for simulation data, provided that sufficiently accurate ISF are available. This requirement is primarily statistical, not methodological. Since ISFs of Brownian particles also exhibit strictly monotonic decay, they can be represented by superpositions of exponentials in the same way as SISFs, making the NNLT directly applicable without any modification.

A second direction is the extension to active Brownian particles (ABP). The main challenge is the construction of a suitable training set that captures the non-monotonic relaxation and the occurrence of non-positive values in the correlation functions. These features may require reconsidering the network architecture, since the treatment of long-time decay in the passive Brownian case relies on activation functions tailored to strictly non-negative, monotonic relaxations. For ABPs, the fluctuations around zero and the oscillatory components may prove problematic. Addressing this will likely require architectural modifications or adapted preprocessing to ensure a stable and accurate inversion of the Laplace transform.

References

[1] S. Hajkowicz, C. Sanderson, S. Karimi, A. Bratanova, and C. Naughtin. "Artificial intelligence adoption in the physical sciences, natural sciences, life sciences, social sciences and the arts and humanities: A bibliometric analysis of research publications from 1960-2021." In: *Technology in Society* 74 (2023). DOI: 10.1016/j.techsoc.2023.102260.

[2] J. Egger et al. "Deep learning–a first meta-survey of selected reviews across scientific disciplines, their commonalities, challenges and research impact." In: *PeerJ Computer Science* 7 (2021). DOI: 10.7717/peerj-cs.773.

[3] Y. Wu, B. Sicard, and S. A. Gadsden. "Physics-informed machine learning: A comprehensive review on applications in anomaly detection and condition monitoring." In: *Expert Systems with Applications* 255 (2024). DOI: 10.1016/j.eswa.2024.124678.

[4] Y. LeCun, Y. Bengio, and G. Hinton. "Deep learning." In: *Nature* 521 (2015). DOI: 10.1038/nature14539.

[5] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[6] K. Hornik, M. Stinchcombe, and H. White. "Multilayer feedforward networks are universal approximators." In: *Neural Networks* 2 (1989). DOI: 10.1016/0893-6080(89)90020-8.

[7] E. Stevens, L. Antiga, and T. Viehmann. *Deep Learning with PyTorch*. Manning Publications, 2020.

[8] R. Jacobs et al. "A practical guide to machine learning interatomic potentials – Status and future." In: *Current Opinion in Solid State and Materials Science* 35 (2025). DOI: https://doi.org/10.1016/j.cossms.2025.101214.

[9] E. Kocer, T. W. Ko, and J. Behler. "Neural Network Potentials: A Concise Overview of Methods." In: *Annual Review of Physical Chemistry* 73 (2022). DOI: 10.1146/annurev-physchem-082720-034254.

[10] J. Behler. "Four Generations of High-Dimensional Neural Network Potentials." In: *Chemical Reviews* 121 (2021). DOI: 10.1021/acs.chemrev.0c00868.

[11] J. Behler and M. Parrinello. “Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces.” In: *Phys. Rev. Lett.* 98 (2007). DOI: 10.1103/PhysRevLett.98.146401.

[12] A. Singraber, T. Morawietz, J. Behler, and C. Dellago. “Parallel Multistream Training of High-Dimensional Neural Network Potentials.” In: *Journal of Chemical Theory and Computation* 15 (2019). DOI: 10.1021/acs.jctc.8b01092.

[13] A. P. Bartók, R. Kondor, and G. Csányi. “On representing chemical environments.” In: *Phys. Rev. B* 87 (2013). DOI: 10.1103/PhysRevB.87.184115.

[14] M. Gastegger, L. Schwiedrzik, M. Bittermann, F. Berzsenyi, and P. Marquetand. “wACSF - Weighted atom-centered symmetry functions as descriptors in machine learning potentials.” In: *The Journal of Chemical Physics* 148 (2018). DOI: 10.1063/1.5019667.

[15] L. Zhang et al. “End-to-end Symmetry Preserving Inter-atomic Potential Energy Model for Finite and Extended Systems.” In: *Neural Information Processing Systems*. 2018.

[16] L. Zhang, J. Han, H. Wang, R. Car, and W. E. “Deep Potential Molecular Dynamics: A Scalable Model with the Accuracy of Quantum Mechanics.” In: *Phys. Rev. Lett.* 120 (2018). DOI: 10.1103/PhysRevLett.120.143001.

[17] S. Batzner et al. “E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials.” In: *Nature Communications* 13 (2022). DOI: 10.1038/s41467-022-29939-5.

[18] D. Marx and J. Hutter. *Ab Initio Molecular Dynamics: Basic Theory and Advanced Methods*. Cambridge University Press, 2009. DOI: 10.1017/CBO9780511609633.

[19] P. Schwerdtfeger and D. J. Wales. “100 Years of the Lennard-Jones Potential.” In: *Journal of Chemical Theory and Computation* 20 (2024). DOI: 10.1021/acs.jctc.4c00135.

[20] J. E. Jones. “On the determination of molecular fields. I. From the variation of the viscosity of a gas with temperature.” In: *Proc. R. Soc. London, Ser. A* 106 (1924). DOI: 10.1098/rspa.1924.0081.

[21] W. Kob and H. C. Andersen. “Testing mode-coupling theory for a supercooled binary Lennard-Jones mixture I: The van Hove correlation function.” In: *Phys. Rev. E* 51 (1995). DOI: 10.1103/PhysRevE.51.4626.

[22] C. Ruscher, J. Baschnagel, and J. Farago. “The Voronoi liquid.” In: *Europhysics Letters* 112 (2016). DOI: 10.1209/0295-5075/112/66003.

[23] C. Ruscher, J. Baschnagel, and J. Farago. “Voronoi glass-forming liquids: A structural study.” In: *Phys. Rev. E* 97 (2018). DOI: 10.1103/PhysRevE.97.032132.

[24] W. Götze. *Complex Dynamics of Glass-Forming Liquids: A Mode-Coupling Theory*. Oxford University Press, 2008. DOI: 10.1093/acprof:oso/9780199235346.001.0001.

[25] W. Götze. “The essentials of the mode-coupling theory for glassy dynamics.” In: *Condensed Matter Physics* 1 (1998). DOI: 10.5488/cmp.1.4.873.

[26] L. M. C. Janssen. “Mode-Coupling Theory of the Glass Transition: A Primer.” In: *Frontiers in Physics* 6 (2018). DOI: 10.3389/fphy.2018.00097.

[27] P. Maréchal, F. Triki, and W. C. Simo Tao Lee. “Regularization of the inverse Laplace transform by mollification.” In: *Inverse Problems* 40 (2024). DOI: 10.1088/1361-6420/ad1609.

[28] C. L. Defreitas and S. J. Kane. “The noise handling properties of the Talbot algorithm for numerically inverting the Laplace transform.” In: *Journal of Algorithms & Computational Technology* 13 (2019). DOI: 10.1177/1748301818797069.

[29] B. Dingfelder. “An improved Talbot method for numerical Laplace transform inversion.” In: *Numerical Algorithms* 68 (2015). DOI: 10.1007/s11075-014-9895-z.

[30] M. P. Allen and D. J. Tildesley. *Computer Simulation of Liquids*. Oxford science publications. Oxford University Press, 2017. DOI: 10.1093/oso/9780198803195.001.0001.

[31] S. A. Hollingsworth and R. O. Dror. “Molecular Dynamics Simulation for All.” In: *Neuron* 99 (2018). DOI: 10.1016/j.neuron.2018.08.011.

[32] L. Verlet. “Computer ”Experiments” on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules.” In: *Phys. Rev.* 159 (1967). DOI: 10.1103/PhysRev.159.98.

[33] W. C. Swope, H. C. Andersen, P. H. Berens, and K. R. Wilson. “A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters.” In: *The Journal of Chemical Physics* 76 (1982). DOI: 10.1063/1.442716.

[34] W. G. Hoover. “Canonical dynamics: Equilibrium phase-space distributions.” In: *Phys. Rev. A* 31 (1985). DOI: 10.1103/PhysRevA.31.1695.

[35] M. E. Tuckerman, J. Alejandre, R. López-Rendón, A. L. Jochim, and G. J. Martyna. “A Liouville-operator derived measure-preserving integrator for molecular dynamics simulations in the isothermal-isobaric ensemble.” In: *Journal of Physics A: Mathematical and General* 39 (2006). DOI: 10.1088/0305-4470/39/19/S18.

[36] A. P. Thompson et al. “LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales.” In: *Comp. Phys. Comm.* 271 (2022). DOI: 10.1016/j.cpc.2021.108171.

[37] *LAMMPS*. https://lammps.org.

[38] G. A. Pavliotis. *Stochastic Processes and Applications*. Springer New York, 2014. DOI: 10.1007/978-1-4939-1323-7.

[39] D. C. Rapaport. *The Art of Molecular Dynamics Simulation*. Cambridge University Press, 2004. DOI: 10.1017/CBO9780511816581.

[40] W. Krauth. *Statistical Mechanics: Algorithms and Computations*. Oxford University Press, 2006. DOI: 10.1093/oso/9780198515357.001.0001.

[41] A. Scala, T. Voigtmann, and C. De Michele. “Event-Driven Brownian Dynamics for Hard Spheres.” In: *The Journal of chemical physics* 126 (2007). DOI: 10.1063/1.2719190.

[42] L. F. Granz. “Event-Driven Brownian Dynamics Simulations of Dense Two-Dimensional Active Brownian Particle Systems.” MA thesis. Universität Düsseldorf, 2020.

[43] J. Reichert, L. F. Granz, and T. Voigtmann. “Transport coefficients in dense active Brownian particle systems: mode-coupling theory and simulation results.” In: *Eur. Phys. J. E Soft Matter* 44 (2021). DOI: 10.1140/epje/s10189-021-00039-4.

[44] J.-P. Hansen and I. R. McDonald. *Theory of Simple Liquids: With Applications to Soft Matter*. Fourth Edition. Academic Press, 2013. DOI: 10.1016/B978-0-12-387032-2.00013-1.

[45] *TikZ: Neural networks*. https://tikz.net/neural_networks.

[46] S. Ruder. *An overview of gradient descent optimization algorithms*. 2017. arXiv: 1609.04747 [cs.LG].

[47] I. Loshchilov and F. Hutter. *Decoupled Weight Decay Regularization*. 2019. arXiv: 1711.05101 [cs.LG].

[48] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. “Learning representations by back-propagating errors.” In: *Nature* 323 (1986). DOI: 10.1038/323533a0.

[49] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind. “Automatic Differentiation in Machine Learning: a Survey.” In: *Journal of Machine Learning Research* 18 (2018).

[50] L. Prechelt. “Early Stopping – But When?” In: *Neural Networks: Tricks of the Trade*. Ed. by G. B. Orr and K.-R. Müller. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998. DOI: 10.1007/3-540-49430-8_3.

[51] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. "Dropout: a simple way to prevent neural networks from overfitting." In: *J. Mach. Learn. Res.* 15 (2014).

[52] A. Krogh and J. A. Hertz. "A simple weight decay can improve generalization." In: Morgan Kaufmann Publishers Inc., 1991.

[53] A. Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library." In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Curran Associates Inc., 2019.

[54] M. Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. 2016. arXiv: 1603.04467 [cs.DC].

[55] G. Doetsch. *Introduction to the Theory and Application of the Laplace Transformation*. Springer Berlin, 1974. DOI: 10.1007/978-3-642-65690-3.

[56] R. N. Bracewell. *The Fourier transform and its applications*. McGraw-Hill, 2000.

[57] E. O. Tuck. "A Simple "Filon-trapezoidal" rule." In: *Mathematics of Computation* 21 (1967). DOI: 10.1090/s0025-5718-67-99892-4.

[58] S. M. Chase and L. D. Fosdick. "An algorithm for Filon quadrature." In: *Commun. ACM* 12 (1969). DOI: 10.1145/363196.363209.

[59] D. Frenkel and B. Smit. *Understanding Molecular Simulation: From Algorithms to Applications*. Academic Press, 2023. DOI: 10.1016/C2009-0-63921-0.

[60] N. Jakse and A. Pasturel. "Interplay between the structure and dynamics in liquid and undercooled boron: An ab initio molecular dynamics simulation study." In: *The Journal of Chemical Physics* 141 (2014). DOI: 10.1063/1.4903452.

[61] J. Sandberg. "Atomic Scale Simulations for Liquid Metals and Alloys: Machine Learning Potentials and Feature Selection." PhD thesis. 2024.

[62] J. Behler. "Atom-centered symmetry functions for constructing high-dimensional neural network potentials." In: *The Journal of Chemical Physics* 134 (2011). DOI: 10.1063/1.3553717.

[63] G. Imbalzano et al. "Automatic selection of atomic fingerprints and reference configurations for machine-learning potentials." In: *The Journal of Chemical Physics* 148 (2018). DOI: 10.1063/1.5024611.

[64] *n2p2*. https://compphysvienna.github.io/n2p2.

[65] A. Singraber, J. Behler, and C. Dellago. "Library-Based LAMMPS Implementation of High-Dimensional Neural Network Potentials." In: *Journal of Chemical Theory and Computation* 15 (2019). DOI: 10.1021/acs.jctc.8b00770.

[66] J. Zeng et al. "DeePMD-kit v2: A software package for deep potential models." In: *The Journal of Chemical Physics* 159 (2023). DOI: 10.1063/5.0155600.

[67] *DeePMD*. https://docs.deepmodeling.com/projects/deepmd.

[68] C. W. Tan et al. *High-performance training and inference for deep equivariant interatomic potentials*. 2025. arXiv: 2504.16068 [physics.comp-ph].

[69] A. Zhu, S. Batzner, A. Musaelian, and B. Kozinsky. "Fast uncertainty estimates in deep learning interatomic potentials." In: *The Journal of Chemical Physics* 158 (2023). DOI: 10.1063/5.0136574.

[70] *NequIP*. https://nequip.readthedocs.io.

[71] M. Geiger and T. Smidt. *e3nn: Euclidean Neural Networks*. 2022. arXiv: 2207.09453 [cs.LG].

[72] B. Kozinsky, A. Musaelian, A. Johansson, and S. Batzner. "Scaling the leading accuracy of deep equivariant models to biomolecular simulations of realistic size." In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. 2023. DOI: 10.1145/3581784.3627041.

[73] J. E. Jones. "On the determination of molecular fields. II. From the equation of state of a gas." In: *Proc. R. Soc. London, Ser. A* 106 (1924). DOI: 10.1098/rspa.1924.0082.

[74] J. E. Jones. "On the determination of molecular fields. III. From crystal measurements and kinetic theory data." In: *Proc. R. Soc. London, Ser. A* 106 (1924). DOI: 10.1098/rspa.1924.0098.

[75] J. E. Jones. "On the atomic fields of helium and neon." In: *Proc. R. Soc. London, Ser. A* 107 (1925). DOI: 10.1098/rspa.1925.0012.

[76] J. E. Jones and A. E. Ingham. "On the calculation of certain crystal potential constants, and on the cubic crystal of least potential energy." In: *Proc. R. Soc. London, Ser. A* 107 (1925). DOI: 10.1098/rspa.1925.0047.

[77] U. R. Pedersen, T. B. Schrøder, and J. C. Dyre. "Phase Diagram of Kob-Andersen-Type Binary Lennard-Jones Mixtures." In: *Phys. Rev. Lett.* 120 (2018). DOI: 10.1103/PhysRevLett.120.165501.

[78] M. Thol, G. Rutkai, R. Span, J. Vrabec, and R. Lustig. "Equation of State for the Lennard-Jones Truncated and Shifted Model Fluid." In: *International Journal of Thermophysics* 36 (2014). DOI: 10.1007/s10765-014-1764-4.

[79] S. Toxvaerd and J. C. Dyre. "Communication: Shifted forces in molecular dynamics." In: *The Journal of Chemical Physics* 134 (2011). DOI: 10.1063/1.3558787.

[80] C. Predescu et al. “Computationally efficient molecular dynamics integrators with improved sampling accuracy.” In: *Molecular Physics* 110 (2012). DOI: 10.1080/00268976.2012.681311.

[81] C. A. Angell, B. E. Richards, and V. Velikov. “Simple glass-forming liquids: their definition, fragilities, and landscape excitation profiles.” In: *Journal of Physics: Condensed Matter* 11 (1999). DOI: 10.1088/0953-8984/11/10A/005.

[82] C. H. Rycroft. “VORO++: A three-dimensional Voronoi cell library in C++.” In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 19 (2009). DOI: 10.1063/1.3215722.

[83] V. Ramasubramani et al. “freud: A Software Suite for High Throughput Analysis of Particle Simulation Data.” In: *Computer Physics Communications* 254 (2020). DOI: 10.1016/j.cpc.2020.107275.

[84] J. L. Meijering. “Interface area, edge length, and number of vertices in crystal aggregates with random nucleation.” In: *Philips Research Reports* 8 (1953).

[85] B. Grünbaum. *Convex Polytopes*. Springer New York, 2003. DOI: 10.1007/978-1-4613-0019-9.

[86] E. V. Podryabinkin and A. V. Shapeev. “Active learning of linearly parametrized interatomic potentials.” In: *Computational Materials Science* 140 (2017). DOI: 10.1016/j.commatsci.2017.08.031.

[87] N. Ravi et al. *Accelerating 3D Deep Learning with PyTorch3D*. 2020. arXiv: 2007.08501 [cs.CV].

[88] A. Okabe, B. Boots, K. Sugihara, S. N. Chiu, and D. G. Kendall. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Wiley, 2000. DOI: 10.1002/9780470317013.

[89] R. Zwanzig. *Nonequilibrium Statistical Mechanics*. Oxford University Press, 2001. DOI: 10.1093/oso/9780195140187.001.0001.

[90] C. Michele, L. Schrack, G. Jung, and F. Thomas. “An Improved Integration Scheme for Mode-Coupling-Theory Equations.” In: *Communications in Computational Physics* 29 (2021). DOI: 10.4208/cicp.OA-2020-0125.

[91] P. Attard. “Chapter 11 - Inhomogeneous Systems.” In: *Thermodynamics and Statistical Mechanics*. Academic Press, 2002. DOI: 10.1016/B978-012066321-7/50011-8.

[92] T. Voigtmann. *mctpy: Mode-Coupling Theory Solver in Python*. https://mctpy.readthedocs.io.

[93] T. Voigtmann. “Mode Coupling Theory of the Glass Transition in Binary Mixtures.” PhD thesis. Technische Universität München, 2003.

[94] J. Reichert. "Transport Coefficients in Dense Active Brownian Particle Systems." PhD thesis. Universität Düsseldorf, 2020.

[95] A. L. Thorneywork et al. "Structure factors in a two-dimensional binary colloidal hard sphere system." In: *Molecular Physics* 116 (2018). DOI: 10.1080/00268976.2018.1492745.

[96] J. J. Arenzon and M. Sellitto. "Microscopic models of mode-coupling theory: The F_{12} scenario." In: *The Journal of Chemical Physics* 137 (2012). DOI: 10.1063/1.4746695.

[97] W. Götze and L. Sjögren. "General Properties of Certain Non-linear Integro-Differential Equations." In: *Journal of Mathematical Analysis and Applications* 195 (1995). DOI: 10.1006/jmaa.1995.1352.

[98] T. Franosch and T. Voigtmann. "Completely Monotone Solutions of the Mode-Coupling Theory for Mixtures." In: *Journal of Statistical Physics* 109 (2002). DOI: 10.1023/A:1019991729106.

[99] T. Franosch and W. Götze. "Relaxation Rate Distributions for Supercooled Liquids." In: *The Journal of Physical Chemistry B* 103 (1999). DOI: 10.1021/jp983412r.

[100] N. Koumakis, J. F. Brady, and G. Petekidis. "Amorphous and ordered states of concentrated hard spheres under oscillatory shear." In: *Journal of Non-Newtonian Fluid Mechanics* 233 (2016). DOI: 10.1016/j.jnnfm.2016.02.004.

[101] J.-L. Barrat and L. Berthier. "Computer simulations of the glass transition and glassy materials." In: *Comptes Rendus. Physique* 24 (2023). DOI: 10.5802/crphys.129.

[102] L. F. Elizondo-Aguilera and T. Voigtmann. "Glass-transition asymptotics in two theories of glassy dynamics: Self-consistent generalized Langevin equation and mode-coupling theory." In: *Phys. Rev. E* 100 (2019). DOI: 10.1103/PhysRevE.100.042601.

[103] H. Teichler. "Evaluation of the memory kernel for fluctuation decay in simulated glass-forming $Ni_{0.5}Zr_{0.5}$ liquids." In: *Phys. Rev. E* 53 (5 1996). DOI: 10.1103/PhysRevE.53.R4287.

[104] A. B. Mutiara and H. Teichler. "Critical temperature T_c and memory kernel in molecular-dynamics-simulated glass-forming $Ni_{0.2}Zr_{0.8}$." In: *Phys. Rev. E* 64 (4 2001). DOI: 10.1103/PhysRevE.64.046133.

[105] M. Kerr Winter, I. Pihlajamaa, V. E. Debets, and L. M. C. Janssen. "A deep learning approach to the measurement of long-lived memory kernels from generalized Langevin dynamics." In: *The Journal of Chemical Physics* 158 (2023). DOI: 10.1063/5.0149764.

[106] G. Szamel and E. Flenner. “Independence of the relaxation of a supercooled fluid from its microscopic dynamics: Need for yet another extension of the mode-coupling theory.” In: *Europhysics Letters* 67 (2004). DOI: 10.1209/epl/i2004-10117-6.

[107] G. Brambilla et al. “Probing the Equilibrium Dynamics of Colloidal Hard Spheres above the Mode-Coupling Glass Transition.” In: *Phys. Rev. Lett.* 102 (2009). DOI: 10.1103/PhysRevLett.102.085703.