University of Milano Bicocca

DEPARTMENT OF MATERIALS SCIENCE
Doctorate in Materials Science and Nanotechnology - XXXVIII Cycle

Accelerating Materials Simulations Across Scales:

Artificial Neural Networks for Atomistic and Continuum Modeling

Candidate:
Andrea Fantasia
Student ID.: 813398

Tutor:
Prof. **Francesco Montalenti**
Supervisor:
Prof. **Roberto Bergamaschini**

PhD project funded by ICSC—Centro Nazionale di Ricerca in High-Performance Computing, Big Data and Quantum Computing, funded by the NextGenerationEU.

Academic year 2024/2025

Contents

Introduction

Over the past decade, the rise of artificial neural network (NN)-based techniques has deeply reshaped the way materials science is approached. From atomistic simulations to continuum modeling and even experimental control, machine learning algorithms and artificial NNs have proven capable of uncovering patterns, surrogating complex physical models, and drastically accelerating simulations.

At the atomic scale, the emergence of machine learning interatomic potentials (MLIPs) represents a major revolution in computational materials science. Traditionally, in large-scale simulations, the modeling of atomic interactions has relied on empirical or semi-empirical potentials such as the Stillinger–Weber [1, 2], Tersoff [3], or Modified Embedded Atom Method (MEAM) [4, 5] potentials. While these classical models are computationally efficient, they suffer from limited transferability and lead to unreliable predictions in metastable phases and kinetic studies [6]. On the other hand, first-principles methods like density functional theory (DFT) offer remarkable accuracy but at a prohibitive computational cost, restricting their use to small systems and short time scales. The introduction of MLIPs closed this gap: they provide an efficient surrogate to DFT-quality potential energy surfaces by learning from *ab initio* datasets [7, 8, 9, 10, 11, 12]. In doing so, they enable simulations of systems orders of magnitude larger and over significantly longer timescales, thus supporting more realistic modeling of complex phenomena such as defect evolution, diffusion, or phase transformations [6, 13].

At the mesoscale and continuum level, materials behavior is often described through models such as the phase field (PF) approach [14]. PF simulations capture microstructural dynamics, including grain growth, phase separation, and pattern formation, within a unified thermodynamic framework. However, the high computational cost associated with solving time-dependent partial differential equations often limits their applicability to long-time or large-scale domains. In this context, deep learning architectures have emerged as powerful alternatives for approx-

imating and replacing traditional numerical solvers [15]. By learning the temporal and spatial evolution of microstructures from representative datasets, these networks can predict system dynamics several orders of magnitude faster than conventional PF solvers [15, 16, 17]. In particular, in recent years, several works have explored convolutional NN–based architectures to reconstruct microstructural evolution sequences, achieving much faster predictions while still preserving essential physical consistency [18, 19, 20, 21].

The main goal of this thesis is to explore and develop artificial NN surrogates aimed at accelerating materials simulations across multiple scales. At the atomistic level, the focus is on developing an MLIP capable of achieving near-DFT accuracy while substantially reducing computational cost. Particular attention is given to designing this potential specifically for studying pressure-induced crystal phase transitions, using germanium as a representative and technologically relevant test system [22]. The primary focus is on demonstrating and evaluating the performance of the potential itself and the technicalities concerning proper sampling of transition-state configurations, rather than providing an exhaustive investigation of germanium phase transitions. For this goal, we used the DeePMD-kit [23, 24], an open-source software package for building MLIPs. It combines a neural-network-based representation of atomic environments, smooth, symmetry-preserving descriptors, and an easy-to-use interface to simulators such as LAMMPS and ASE [25].

At the continuum scale, the objective is to develop deep-learning-based surrogate models that approximate PF evolutions with high fidelity and significant speed-up. In particular, we aim to build a convolutional recurrent NN architecture, using PyTorch [26], aimed at accelerating PF simulations of spinodal decomposition in coherently strained alloy systems. This system was selected because it naturally generates complex and diverse patterns depending on a governing parameter, making it an ideal case study for evaluating the capabilities of the proposed architecture. The emphasis, however, is on the generality of the framework itself, which is versatile and applicable to a broader class of continuum models.

The thesis is outlined as follows:

Part I introduces the general framework of modern data-driven surrogate modeling. Chapter 1 provides an overview of surrogate modeling, discussing its motivation, strategic sampling approaches, and general principles for model development and optimization. Chapter 2 focuses on feed-forward NNs as surrogate models, covering the fundamentals of ar-

tificial NNs, training via loss minimization, and practical considerations relevant to their application.

Part II is devoted to atomistic modeling. Chapter 3 presents the theoretical background and computational methods for atomistic simulations, providing a brief explanation of density functional theory and the application of molecular dynamics and saddle-search approaches for generating datasets for MLIPs. Particular emphasis is given to discussing those tools specifically suited for exploring and predicting solid-state transition configurations, such as the solid-state nudged elastic band and the solid-state dimer methods, extensively used in our work. Chapter 4 reviews the main classes of MLIPs, including a brief historical timeline, a more in-depth discussion of the DeePMD framework we used, and a mention of recent developments in the field. In Chapter 5, the results of this part of the thesis work are discussed: the development of our MLIP for germanium is presented, detailing the model architecture, training strategy based on iterative active learning, benchmarking and validation, as well as applications to the investigation of some transition pathways and pressure-dependent kinetics.

Part III addresses continuum modeling. Chapter 6 introduces the theoretical background and numerical methods for continuum-scale simulations, specifically focusing on PF modeling of coherently strained spinodal decomposition and details on the dataset generation for our model. Then, deep learning architectures for learning continuum evolution, including convolutional, recurrent, and other emerging approaches, are discussed. In Chapter 7, the deep-learning framework we developed for accelerating simulations of coherent spinodal decomposition is presented, providing details on the adopted convolutional recurrent NN architecture, training and validation procedures, and results from several tests aimed at supporting the model's robustness in temporal and domain-size generalization, among other aspects.

Finally, the thesis concludes with a summary of the main results and perspectives for future research.

PART I
Foundations and Methodologies of Modern Data-Driven Surrogate Modeling

Chapter 1

Introduction to Surrogate Modeling

As the objective of this thesis is the acceleration of materials simulations, this section introduces the key general concepts and practical techniques necessary to address this challenge. Our strategy relies on developing surrogate models based on artificial neural networks (NNs), which can approximate the results of computationally expensive simulations using suitably generated training data and enabling much faster predictions.

In this first part of the thesis, we provide a concise overview of surrogate modeling, outline the main technical aspects of constructing such models, and review both standard and state-of-the-art methods and tools, including NNs. We also present a justification for choosing NNs over alternative approaches, discussing their advantages and limitations.

1.1 Overview and Motivation

The application of surrogate modeling spans multiple use cases in science and engineering. One of the most direct applications, which is our focus, involves accelerating expensive simulations. To mention a few applications, beyond simple interpolation or curve fitting, surrogate models can be useful as calibration mechanisms in multi-fidelity frameworks, bridging between fast but approximate models and slower, more accurate reference methods. They also provide robustness against noisy or missing data, smoothing computational artifacts arising from discretization schemes or incomplete convergence. Furthermore, surrogates facilitate sensitivity analysis, revealing which design variables exert the strongest influence on quantities of interest and enabling visualization of high-dimensional

response surfaces [27].

When data is highly noisy, surrogate modeling overlaps with statistical learning. The two fields share many methods, with statistical learning emphasizing inference and prediction of real-world noisy data and therefore tends to rely more explicitly on regularization than surrogate modeling in low-noise, simulation-driven settings.

The construction and deployment of accurate surrogate models, nevertheless, demands careful attention to several interconnected methodological considerations. First, the sampling strategy must ensure that the training data adequately cover the relevant input space, capturing critical features such as, for the applications discussed in this Thesis, phase boundaries, instabilities, and rare events without requiring excessive parametric sweeps. For this reason, there are multiple techniques (here mentioned and discussed later), such as active learning and other adaptive refinement methods, that enable efficient exploration of high-dimensional parameter spaces. Second, the choice of surrogate architecture, whether Gaussian processes, polynomial response surfaces, or modern deep neural networks, must match the complexity and structure of the underlying physical system. As a rule of thumb, a simpler model is generally preferred over a more complex one, provided they achieve reasonably similar performance in terms of accuracy and speed (you shouldn't use a hammer to crack a walnut). It should be standard practice to first develop a simple baseline model before exploring more sophisticated approaches, and to use the baseline's performance as a reference point for judging whether the added complexity is justified. Third, surrogate models must be rigorously validated and tested on unseen data to assess generalization performance and identify regions of parameter space where predictions might be unreliable [28, 29, 30].

A particularly critical challenge arises from the curse of dimensionality: as the number of input parameters increases, the volume of parameter space grows exponentially, and the density of training data required for adequate coverage becomes prohibitively expensive. In materials modeling, simulation outputs may depend on numerous material properties, different atomic configurations, geometric parameters, and other conditions. In general, strategies to mitigate dimensionality include dimensionality reduction techniques, active subspace methods, and sequential experimental designs that iteratively refine the surrogate in regions of greatest interest [27, 28, 29]. For our use case, more targeted strategies are discussed in the dedicated sections.

1.2 Strategic Sampling

The accuracy and generalization of a data-driven surrogate model depend heavily on the distribution of the training samples across the sample domain. As a simple example, for instance, clustering points in certain subdomains forces the surrogate to extrapolate unreliably elsewhere. So, already one can imagine that as a good rule of thumb, the sample distribution should be such as not to leave any empty holes in the sampled domain. Strategic sampling methods address this challenge by systematically selecting training points that maximize information gain while respecting computational budget constraints [31, 32]. A well-designed sampling plan should exhibit space-filling properties, ensuring uniform model accuracy throughout the domain by avoiding regions where predictions must rely on distant extrapolation [33].

Basic Sampling Techniques

The two simplest, yet in most case effective, widespread methods are grid sampling and random sampling. In grid sampling, the parameter space is discretized uniformly along each dimension, indeed forming a grid. In practice, all combinations of parameter values are systematically evaluated. This method has excellent coverage, and it's simple and easy to visualize, but it can lead to aliasing effects in some cases. In random sampling, on the other hand, samples are drawn uniformly at random from the parameter space. This is still simple to implement and avoids structured sampling artifacts, but, depending on the seed, coverage might be uneven, and large gaps may occur, especially if the sample size is quite small.

Another more thoughtful technique is Latin hypercube (LHC) sampling. With this method, the parameter space is divided into equal-probability intervals in each dimension. One point is sampled from each interval, ensuring that each dimension is uniformly sampled without repetition. This method leads to much better uniformity than pure random sampling and typically requires far fewer samples than grid sampling. However, random LHCs may still exhibit poor space-filling characteristics in higher dimensions despite satisfying the projection criterion: points might cluster along diagonals or leave large voids unexplored. To address this, optimized LHC variants are constructed by incorporating distance-based criteria that explicitly promote uniform point dispersion [27, 31, 32, 33].

Sobol and Halton sequences sampling

Sobol and Halton sequences are a type of quasi-random (low discrepancy) sequence, meaning they are designed to produce points that cover a multidimensional space more evenly than simple random sampling. Unlike random sampling, where clustering and gaps may appear, these sequences strategically place points to minimize these irregularities.

Points of the sequence are generated by a deterministic algorithm: although they look random, running the sequence from the same seed always produces the same points. The main advantage is their exceptional uniformity in filling the space, reducing regions of oversampling or undersampling. We do not delve into the details of the implementations, but the takeaway is that Sobol's construction offers unparalleled uniformity in moderate to high dimensions, making it highly desirable where efficiency matters. Halton is easier to implement but less robust in very high dimensions. Compared to grid, random, and LHS methods, their superior convergence and uniformity offer significant practical advantages. The main limitation of these quasi-random sampling techniques is that they are not always optimal for some statistical or stochastic analysis where true randomness is required [34, 35].

Adaptive Sampling and Sequential Refinement

While one-shot sampling methods can provide excellent initial coverage, surrogate model accuracy can often be substantially improved through adaptive sampling strategies. Adaptive sampling consists in the iterative selection of new training points based on information extracted from the current surrogate. Adaptive methods are particularly valuable when the response surface exhibits localized features or sharp gradients that initial sampling may inadequately capture [36, 37, 38, 39].

1.3 Development and Optimization of Surrogate Models

The construction of a surrogate model involves transforming sampled input-output pairs into a predictive function capable of generalizing to unseen parameter configurations. This process, fundamentally an exercise in supervised learning, requires navigating a vast space of candidate functions to identify one that not only reproduces training observations effectively but also possibly captures the underlying physics governing the expensive simulator's behavior. The central challenge lies in balancing

model flexibility (the capacity to represent complex response surfaces) against the risk of overfitting, wherein the surrogate learns the training data but is unable to generalize with satisfactory predictive accuracy on new points [27].

The Surrogate Modeling Workflow

Surrogate model construction typically proceeds through three sequential stages. Stage One encompasses data preparation and model architecture selection, building upon the sampling strategies discussed previously. For most applications, input variables are conventionally scaled to the unit hypercube $[0, 1]^k$ to facilitate numerical stability. The choice of surrogate architecture, whether a simple linear regression or a more complicated NN architecture, must reflect prior knowledge about the expected complexity of the response surface [27].

Stage Two addresses parameter training: given a fixed model structure $\hat{f}(\mathbf{x}; \theta)$, the parameters θ are tuned to optimally fit the observed data $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{n}$. The estimation criterium for traditional models is the maximum likelihood estimation, which seeks parameters that maximize the probability of observing the training data, assuming a specific noise model. For Gaussian-distributed errors with standard deviation σ, the likelihood becomes:

$$\mathcal{L} = \frac{1}{(2\pi\sigma^2)^{n/2}} \prod_{i=1}^{n} \exp\left(-\frac{(y^{(i)} - \hat{f}(\mathbf{x}^{(i)}; \theta))^2}{2\sigma^2} \right) \tag{1.1}$$

Maximizing this likelihood, or equivalently, minimizing its negative logarithm, yields the ordinary least squares criterion when σ is constant:

$$\min_{\theta} \sum_{i=1}^{n} \left(y^{(i)} - \hat{f}(\mathbf{x}^{(i)}; \theta) \right)^2 \tag{1.2}$$

Since this scenario encompasses most of the practical cases, in simple terms, the above paragraph translates into: often, the objective of model training is to minimize the root mean square error (RMSE) of the predicted outputs.

Notice that this evaluation of the RMSE on the training data alone does not tell us how well our model will generalize to new, unseen data. Especially when we train a complex model, it might just memorize the training data: we could get a very low RMSE on that data, even if the model performs poorly on new cases. To have a better control on

this behaviour, a standard approach is to use cross-validation. In cross-validation the training data is partitioned into q subsets, iteratively withholding each subset for validation while training on the remaining $q-1$ folds. The cross-validation error estimate is computed as:

$$\epsilon_{\text{cv}}(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)} - \hat{f}_{-\kappa^{(i)}}(\mathbf{x}^{(i)};\theta)\right)^2 \tag{1.3}$$

where $\hat{f}_{-\kappa^{(i)}}$ denotes the predictor trained excluding the subset $\kappa^{(i)}$ containing observation i. Typical chosen values are $q = 5$ or $q = 10$ as they represent practical compromises between computational cost and estimation variance. Repeating the cross-validation procedure across multiple hyperparameter choices provides a systematic, data-driven approach to identifying the best hyperparameters of the model.

Finally, Stage Three focuses on testing the trained model on data that was never used during either training or hyperparameter tuning. This held-out test set provides an unbiased estimate of how well the surrogate will perform on truly new cases. When sufficient observations are available, it is good practice to reserve roughly 20–25% of the dataset exclusively for this final evaluation, keeping it completely isolated until all modeling and tuning decisions have been finalized.

Next, we review some of the most widely used surrogate models in materials science and engineering, including polynomial response surfaces, radial basis function models, Gaussian process regression, and others that balance accuracy with computational efficiency. It's worth mentioning that there are powerful tree-ensemble regressors, such as random forests, gradient boosting, and XGBoost, but they are not our focus here because they yield piecewise-constant predictions rather than a smooth continuous mapping over the input space, which is the type of surrogate we are interested in.

Polynomial Response Surfaces

Polynomial models represent the simplest class of surrogate, expressing the response as a truncated Taylor series expansion [27]:

$$\hat{f}(\mathbf{x};\mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \cdots + w_m x^m = \sum_{i=0}^{m} w_i x^i \tag{1.4}$$

This holds for univariate problems. More generally, a polynomial response surface is a surrogate model in which an expensive response $f(\mathbf{x})$ is approximated by a linear combination of multivariate polynomial

basis functions. Given input–output samples $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{n}$, we select a set of basis functions $\{\phi_j(\mathbf{x})\}_{j=1}^{n_b}$ (e.g., monomials and interaction terms) and write the surrogate as $\hat{f}(\mathbf{x}) = \sum_{j=1}^{n_b} w_j \phi_j(\mathbf{x})$. Collecting all basis functions evaluated at all sample points forms the Vandermonde (design) matrix Φ, with entries $\Phi_{ij} = \phi_j(\mathbf{x}^{(i)})$. The coefficient vector $\mathbf{w}$ is then obtained by linear least squares, which in maximum likelihood form under Gaussian noise gives

$$\mathbf{w} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{y} = \Phi^{+} \mathbf{y}, \tag{1.5}$$

where Φ^{+} is the Moore–Penrose pseudo-inverse [40]. This construction yields a closed-form, globally defined surrogate that is cheap to evaluate and differentiable.

The polynomial order m governs model flexibility: higher orders can capture increasingly complex landscapes but risk overfitting noise. Cross-validation provides an effective means to select m across candidate orders and choose the value yielding the minimum estimated prediction error. Polynomial surrogates offer algebraic simplicity and interpretability, making them valuable when physical insight suggests low-order relationships or when symbolic manipulation is required downstream. However, their global rigidity limits applicability to highly nonlinear or multimodal response surfaces.

Polynomial Spline Regression

Polynomial spline regression addresses the limitations of global polynomial regression by partitioning the predictor domain into intervals and fitting separate low-degree polynomials within each interval, connected smoothly at boundaries called knots. This procedure yields a more flexible, piecewise model that avoids the instability of high-degree global polynomials. The model approximates the unknown function as a linear combination of spline basis functions $B_j(x)$, so that $\sum_{j=1}^{d} \gamma_j B_j(x_i)$ represents the fitted value at x_i, where $B_j(\cdot)$ are B-spline basis functions and γ_j are their coefficients to be estimated. B-splines are piecewise polynomial functions defined on a set of knot locations that partition the domain of x. They have local support, meaning each basis function is nonzero only over a limited interval. Typically, the coefficients γ_j are estimated by minimizing a penalized least squares criterion of the form:

$$\mathcal{L}_\lambda(\gamma) = \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{d} \gamma_j B_j(z_i) \right)^2 + \lambda \sum_{j=l+2}^{d} \gamma_j^2, \tag{1.6}$$

where $\lambda \geq 0$ is a smoothing parameter controlling the penalty term. In this way, we can have a trade-off between closeness to the data and smoothness of the fitted curve: larger λ produces smoother, less wiggly fits. In practice, the knot positions and the number of knots are not chosen manually. Procedures such as cross-validation or similar criteria are commonly used to select λ and adjust the knot configuration, so that the resulting spline balances flexibility and smoothness without overfitting.

Radial Basis Function Models

Radial basis function (RBF) models approximate an expensive response $f(\mathbf{x})$ as a weighted sum of localized, radially symmetric basis functions centered at prescribed points in the input space. Given centers $\{\mathbf{c}^{(i)}\}_{i=1}^{n_c}$, the surrogate takes the form

$$\hat{f}(\mathbf{x}, \mathbf{w}) = \mu + \sum_{i=1}^{n_c} w_i \, \phi(\|\mathbf{x} - \mathbf{c}^{(i)}\|), \tag{1.7}$$

where $\phi(\cdot)$ is a chosen radial kernel (e.g., linear, cubic, thin plate spline, Gaussian, multiquadric, or inverse multiquadric) and $\|\cdot\|$ is the Euclidean norm. Ideally, the choice of the radial kernel should be made aiming at getting the best fit with fewer parameters. From here, the construction is not too different from polynomial response surfaces: one selects basis functions (via the centers and ϕ), assembles the RBF matrix $\mathbf{\Psi}$ with entries $\Psi_{ij} = \phi(\|\mathbf{x}^{(i)} - \mathbf{c}^{(j)}\|)$, and solves a linear system (or regularized least squares) to obtain the weights $\mathbf{w}$. For interpolation, one can simply use one center per data point ($n_c = n$), whereas for noisy regression, a smaller set of centers ($n_c < n$) is employed. When Gaussian or inverse multiquadric kernels are used, $\mathbf{\Psi}$ is symmetric positive definite, which permits an efficient and numerically stable solution of the coefficient system via Cholesky factorization, while the localized, distance-based basis functions enable flexible approximation of highly nonlinear responses.

Gaussian Process Regression

Kriging or Gaussian process regression (GPR) represents a particularly sophisticated RBF variant that has become standard in surrogate-based optimization, and, as we will discuss in Sec. 4.1, it is at the basis of a powerful class of machine learning interatomic potentials, such as GAP [41]. GPR models the unknown response as a random function with a Gaussian process prior, written as

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')). \tag{1.8}$$

This means that for any finite set of inputs, the corresponding function values are modeled as jointly Gaussian, with mean given by the mean function $m(\mathbf{x})$ and covariance determined by the kernel $k(\mathbf{x}, \mathbf{x}')$. The kernel acts as a similarity measure: points that are "close" according to k are expected to have strongly correlated function values, which controls both the smoothness of the surrogate and how information propagates from observed data to new locations. The usual kernel choice is a parametric Gaussian kernel:

$$k(x, x') = \sigma_f^2 \exp\left(-\frac{(x - x')^2}{2l^2}\right) \tag{1.9}$$

with σ_f^2 signal variance and l characteristic length scale. Kernel hyperparameters are inferred by maximizing the marginal likelihood of the data, allowing the model to adapt automatically to the complexity of the underlying response.

In the noise-free case, GPR reduces to an interpolant: the posterior mean exactly matches the observed data at the training points, and the predictive variance vanishes there, so the surrogate passes through all samples while still providing uncertainty estimates away from them. Following the above assumptions, given data $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^n$, for any test input x_*, the model prediction will be:

$$\hat{f}(\mathbf{x}_* \mid \{(\mathbf{x}^{(i)}, y^{(i)})\}) \sim \mathcal{N}\big(\mu(\mathbf{x}_*), \sigma^2(\mathbf{x}_*)\big). \tag{1.10}$$

The mean and variance are given by

$$\mu(\mathbf{x}_*) = k_*^T K_y^{-1} \mathbf{y}, \quad \sigma^2(\mathbf{x}_*) = k(\mathbf{x}_*, \mathbf{x}_*) - k_*^T K_y^{-1} k_*, \tag{1.11}$$

where

$$k_* = [k(\mathbf{x}_*, \mathbf{x}^{(1)}), \ldots, k(\mathbf{x}_*, \mathbf{x}^{(n)})]^T, \quad K_y = K + \sigma_n^2 I, \tag{1.12}$$

with elements of K given by $K_{ij} = k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$, and σ_n^2 being the variance of the eventual noise in $\mathbf{y}$ data.

Constructing the GP surrogate thus involves forming the $n \times n$ covariance matrix K_y and solving a linear system to find the vector of weights of the form $\boldsymbol{\alpha} = K_y^{-1}\mathbf{y}$ (often via Cholesky factorization), leading to a

computational cost of $O(n^3)$ that limits scalability. Despite this limitation, GPR remains highly attractive for surrogate-based optimization as it provides both accurate interpolation in the noise-free regime and flexible regression with useful uncertainty quantification.

Chapter 2

Feed-forward Neural Networks as Surrogate Models

2.1 Introduction to Artificial Neural Networks

Artificial neural networks [42] emerge as a new powerful tool that can be used, in our case, to build surrogate models suited to high-dimensional, large-scale regression tasks. NNs are universal approximators: given sufficient architecture complexity, number of parameters, and appropriate training, they can, in principle, approximate any continuous function on compact domains to arbitrary precision [43]. This flexibility is not their only strength. NNs' linear computational scaling in training set size and their ability to leverage parallel computing architectures (modern fast graphics processing units) make them particularly attractive for data-intensive applications in any sector [44].

However, this expressiveness comes at a cost: NNs often require very careful regularization to prevent overfitting, their training landscape is nonconvex (rendering optimization challenging), and they traditionally provide no principled measure of prediction uncertainty. Despite these challenges, the last two decades have witnessed incredible growth in deep learning applications, driven by advances in optimization algorithms, regularization techniques, hardware acceleration, and novel architectures. In this section, we introduce the theoretical foundations of NN surrogates while maintaining focus on applications to materials simulations.

Feedforward Neural Network Architecture

A feedforward neural network, often referred to as a multilayer perceptron, is composed of a sequence of layers of interconnected neurons forming a so-called directed acyclic computational graph. Information propagates strictly forward: inputs are presented at the input layer, transformed through one or more hidden layers, and finally mapped to outputs at the output layer, with no recurrent or lateral connections that would create cycles. For a surrogate modeling task with k input parameters and a scalar output, the architecture is defined by three main components:

- An input layer with k neurons, each corresponding to one design variable in the vector $\mathbf{x} = (x_1, \ldots, x_k)^T$
- A stack of L hidden layers, where hidden layer l contains $n_h^{(l)}$ neurons
- An output layer with a single neuron that produces the model prediction $\hat{f}(\mathbf{x})$.

Neurons in consecutive layers are fully connected, so every neuron in layer l receives the activations (a certain signal) from all neurons in layer $l-1$. The transformation performed by layer l can be written compactly:

$$\mathbf{a}^{(l)} = \sigma\big(\mathbf{W}^{(l)}\mathbf{a}^{(l-1)} + \mathbf{b}^{(l)}\big), \tag{2.1}$$

where $\mathbf{a}^{(l-1)}$ is the vector of activations from the previous layer, $\mathbf{W}^{(l)}$ is a weight matrix of size $n_h^{(l)} \times n_h^{(l-1)}$ that linearly combines those inputs, and $\mathbf{b}^{(l)}$ is a bias vector of length $n_h^{(l)}$ that shifts the transformation. The input to the first hidden layer is identified as $\mathbf{a}^{(0)} = \mathbf{x}$; the so-called activation function, here denoted as $\sigma(\cdot)$, is applied element-wise to introduce nonlinearity. By composing these layer-by-layer transformations, the NN defines a parametric mapping from the input space to the output response that is trained to approximate the target function.

Activation functions

Activation functions introduce the nonlinearity required for a NN to approximate complex input–output relationships. Without activation functions, it's easy to check that the entire network would effectively collapse to a single linear map regardless of depth. Common choices include:

- The hyperbolic tangent, $\sigma(z) = \tanh(z)$, which is an S-shaped curve, zero-centered and maps to $(-1, 1)$
- The Rectified Linear Unit (ReLU) [45], $\sigma(z) = \max(0, z)$, which is piecewise linear and computationally cheap.

Also, there are some variants that refine ReLU's behavior, such as Leaky ReLU [46], which assigns a small nonzero slope to negative inputs, and Exponential Linear Units (ELU) [47], which introduces a smooth negative branch. These and more variants (e.g., GELU [48], Softplus [49]) were introduced to address a key limitation of the standard ReLU activation function: the "dying ReLU" problem, where neurons "become dead" during training and stop learning because their gradients become zero for negative values. Also, another avoided issue is that by using the plain ReLU, the output of the NN would be piece-wise linear, while often a smooth response curve is preferred.

For a network with L layers, the overall forward mapping from input $\mathbf{x}$ to scalar output $\hat{f}(\mathbf{x}; \theta)$ can be written as a nested composition of affine transformations and activation functions,

$$\hat{f}(\mathbf{x}; \theta) = \mathbf{W}^{(L)}\, \sigma\big(\mathbf{W}^{(L-1)}\, \sigma(\cdots \sigma(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}) + \mathbf{b}^{(L-1)})\big) + \mathbf{b}^{(L)}, \quad (2.2)$$

where θ denotes the collection of all weight matrices and bias vectors. The total number of learnable parameters is $\sum_{l=1}^{L} n_h^{(l)}(n_h^{(l-1)}+1)$, based on the product of incoming connections and biases per neuron, and typically ranges from hundreds to millions in practical architectures. Notably, the largest NN-based architecture at the time of this thesis (end of 2025) is a GPT model with $\sim$1.8 trillion parameters.

2.2 Training via loss minimization

The surrogate parameters θ are estimated by minimizing a loss function quantifying discrepancy between predictions and training targets, usually using the MSE:

$$\mathcal{L}(\theta) = \frac{1}{n} \sum_{i=1}^{n} (y^{(i)} - \hat{f}(\mathbf{x}^{(i)}; \theta))^2 \quad (2.3)$$

Mind that this is only one of many alternative formulations depending on the specific task, whether it be regression, classification, quantile regression, or other physics-informed objectives. For regression tasks,

the two other common choices are the MAE, which is more robust to outliers than MSE, and the Huber loss [50], which behaves like MSE near zero error and like MAE for large errors, balancing smoothness and robustness.

Backpropagation

Minimization proceeds via gradient-based optimization, wherein the loss gradient with respect to the NN's parameters is computed via a key, simple yet powerful algorithm known as backpropagation. The goal of this algorithm is to tweak each weight a little so that next time the loss function is computed, the error is smaller. In practical terms, for each weight we want to know: "If we nudge this weight slightly, does the loss go up or down, and by how much?" That "how much?" is given by the gradient $\partial\mathcal{L}/\partial\theta$, computed for each parameter θ. Backpropagation applies the chain rule to efficiently compute gradients layer-by-layer, from the output layer backward to the input layer:

$$\frac{\partial\mathcal{L}}{\partial\mathbf{W}^{(l)}} = \frac{\partial\mathcal{L}}{\partial\mathbf{a}^{(l)}}\frac{\partial\mathbf{a}^{(l)}}{\partial\mathbf{W}^{(l)}} \tag{2.4}$$

with $\partial\mathcal{L}/\partial\mathbf{a}^{(l)}$ computed recursively from layer L back to layer 1.

Because intermediate activations produced during the forward pass are stored and reused, and each layer depends only on information from adjacent layers, the overall cost of computing all gradients scales linearly with the number of parameters. This efficiency in parameter optimization, combined with the inherent flexibility of neural networks, is the key reason for their practical power and widespread success. Now that we have established how the gradients of the loss function are computed, we need to choose a suitable gradient-based optimizer to minimize the loss.

Optimizers

Given parameters θ_t at iteration t, an optimizer computes an update $\Delta\theta_t$ based on the (typically stochastic) gradient of the loss, giving $\theta_{t+1} = \theta_t + \Delta\theta_t$. In so-called full-batch gradient descent, $\Delta\theta_t$ is proportional to the gradient of the loss over the entire training set. For large datasets, however, this is computationally expensive, and stochastic gradient descent (SGD) with mini-batches is employed [51]. At each iteration, the gradient is estimated on a subset B_t of the training data,

$$g_t = \frac{1}{|B_t|}\sum_{i\in B_t}\nabla_\theta\mathcal{L}(\mathbf{x}^{(i)};\theta_t|y^{(i)}), \tag{2.5}$$

and parameters are updated as

$$\theta_{t+1} = \theta_t - \eta\, g_t, \tag{2.6}$$

where $\eta > 0$ is the learning rate. The batch size $|B_t|$ controls a trade-off between computational efficiency and gradient noise: small batches yield noisy but frequent updates that can improve generalization, whereas large batches produce smoother gradients but may converge more easily to suboptimal minima.

Several extensions of SGD have been proposed to accelerate convergence and improve stability. SGD with momentum introduces a velocity variable v_t that accumulates an exponential moving average of past gradients, smoothing oscillations and promoting progress along low-curvature directions:

$$v_{t+1} = \mu v_t - \eta\, g_t, \qquad \theta_{t+1} = \theta_t + v_{t+1}, \tag{2.7}$$

where μ is the momentum coefficient. Intuitively, one can picture this algorithm to be similar to simulating a ball rolling down a hilly landscape, not just moving exactly along the steepest descent, but conserving some momentum along the way. This also helps avoid getting stuck in shallow suboptimal minima.

Another class of algorithms is adaptive learning-rate methods. They further rescale updates per parameter based on the history of gradients. AdaGrad [52] maintains an accumulated squared gradient r_t and uses a parameter-wise effective step size $\eta/(\sqrt{r_{t+1}} + \epsilon)$ (ϵ small positive constant), which is beneficial for unbalanced gradients but can lead to overly small learning rates as r_t grows. RMSProp addresses this by replacing the cumulative sum with an exponential moving average,

$$r_{t+1} = \rho r_t + (1-\rho) g_t^2, \qquad \theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{r_{t+1}} + \epsilon}\, g_t, \tag{2.8}$$

thus maintaining a stable scale of updates throughout training.

Another useful method is Adam (Adaptive Moment Estimation) [53], which combines momentum and RMSProp-style adaptation by maintaining exponential moving averages of both first and second moments of the gradients. At iteration t, the moment estimates and their bias-corrected forms are:

$$m_{t+1} = \beta_1 m_t + (1-\beta_1) g_t, \qquad v_{t+1} = \beta_2 v_t + (1-\beta_2) g_t^2, \tag{2.9}$$

$$\hat{m}_{t+1} = m_{t+1}/(1-\beta_1), \qquad \hat{v}_{t+1} = v_{t+1}/(1-\beta_2). \tag{2.10}$$

where m_t is the first moment (mean gradients), and v_t is the second moment (mean of squared gradients), both starting from $m_0, v_0 = 0$. The update rule is then

$$\theta_{t+1} = \theta_t - \eta \frac{\hat{m}_{t+1}}{\sqrt{\hat{v}_{t+1}} + \epsilon}. \tag{2.11}$$

Adam and its variants (e.g., AdamW with decoupled weight decay, and Nadam with Nesterov-style momentum) have become widely used defaults in deep learning due to their robustness to noisy and heterogeneous gradients and relatively low sensitivity to initial learning-rate choices.

In practice, Adam is often the default optimizer when one is unsure where to begin. However, despite its robustness, there is no guarantee that Adam will yield the best-performing model [54]. When aiming for a highly competitive and accurate model (and when computational resources permit), it is generally advisable to experiment with multiple optimizers, various hyperparameter settings, and different parameter initialization seeds. A further consideration is that, in NN training, the loss landscape is typically highly non-convex, leading different optimizers, hyperparameters, and initialization seeds to converge to distinct local minima with significantly varying generalization performance.

2.3 Additional practical considerations

As already mentioned, Neural networks' high flexibility with architectures often made up by tens of thousands of parameters renders them susceptible to overfitting: fitting the training data accurately but failing to learn generalizable patterns. While preparing a NN model, it's standard procedure to divide the sampled data into training and validation sets, with a usual split being 80% : 20%. Having a separate validation set allows us to monitor the model's behavior during training and mitigate overfitting. Overfitting typically appears when the training loss continues to decrease while the validation loss increases, signaling poor generalization to unseen data. Several complementary regularization strategies can help mitigate this problem.

Regularization terms in the loss function

A common way to regularize neural networks is to modify the loss function itself by adding a penalty that discourages overly complex parameter configurations. In general, we replace the original loss $\mathcal{L}(\theta)$ with a regularized one

$$\mathcal{L}_{\text{reg}}(\theta) = \mathcal{L}(\theta) + \Omega(\theta), \tag{2.12}$$

where $\Omega(\theta)$ is a penalty term, and its strength is controlled by one or more hyperparameters.

There are three common choices here. One is L2 (Ridge) regularization, which consists of a term $\Omega(\theta) = \lambda \sum_j \theta_j^2$. This penalizes large weights and effectively pulls parameters toward zero, which tends to produce smoother, less sensitive mappings and better generalization.

A second closely related variant is L1 (Lasso) regularization, which consists of a term $\Omega(\theta) = \lambda \sum_j |\theta_j|$. The absolute-value penalty encourages many parameters to become exactly zero, inducing sparsity in the weights. In neural networks, this can act as a kind of feature selection or pruning: only a subset of connections remains active.

Lastly, we can use Elastic Net regularization, which combines L1 and L2 terms:

$$\mathcal{L}_{\text{reg}}(\theta) = \mathcal{L}(\theta) + \lambda \left(\alpha \sum_j |\theta_j| + (1-\alpha) \sum_j \theta_j^2 \right). \tag{2.13}$$

In practice, this can yield models that are both compact and numerically well behaved.

In all these cases, the optimizer now minimizes $\mathcal{L}_{\text{reg}}(\theta)$ instead of $\mathcal{L}(\theta)$. The gradients simply gain an extra term from the penalty, so regularization integrates seamlessly with backpropagation and the chosen optimizer. By tuning the regularization strength λ using the validation set, one can control the bias–variance trade-off: large penalties increase bias but often substantially reduce variance and overfitting.

Dropout layers

Another regularization approach tailored specifically to neural network models is the use of dropout layers [55, 56, 57]. Dropout is employed as a stochastic regularization mechanism that reduces overfitting by randomly deactivating a certain number of neurons during training. For a given layer with activations h, dropout samples a binary mask $m \in \{0,1\}^d$ and forms the modified activations

$$\tilde{h}_j = \frac{m_j}{1 - p_{\text{drop}}} h_j, \tag{2.14}$$

where p_{drop} denotes the dropout rate. In practice, each neuron is deactivated with probability p_{drop}, and the output values of the remaining active neurons are rescaled. This formulation ensures that the expected activation is preserved, $\mathbb{E}[\tilde{h}_j] = h_j$, while injecting this kind of noise during training. At inference time, dropout is disabled, and the full deterministic network is used, reusing the weights that have been shaped under this stochastic training regime (without the rescaling factor). Intuitively, dropout can be viewed as a mechanism that discourages co-adaptation, forcing each neuron to learn features that remain useful across many random subnetworks.

Dropout is, in many cases, useful also beyond regularization, as it can be exploited as an implicit ensemble method. Indeed, maintaining dropout at test time and performing multiple stochastic forward passes (Monte Carlo dropout) provides an approximate Bayesian treatment of model uncertainty, enabling empirical estimation of predictive means and variances via repeated sampling, which can be used to construct approximate confidence intervals in surrogate modeling applications.

Batch normalization

Another problem that may arise during the training of deep neural networks is internal covariance shift: the distribution of neuron activations may drift, because weight updates in earlier layers change the inputs seen by later layers, amplifying small shifts across the network and causing unstable, slow training. Batch normalization is a widely used technique that stabilizes and accelerates the training of neural networks by normalizing the outputs of each layer across a mini-batch [58]. During training, it adjusts the activations of each layer so that they have a more consistent distribution (zero mean and unit variance), thereby reducing the problem of internal covariate shift. After training, batch normalization applies learned scaling and offset parameters. This approach enables typically higher learning rates, faster convergence, and improved model stability. After the model is trained, during inference, batch normalization instead uses learned parameters of means and variances, ensuring consistent predictions even when processing one sample at a time.

Hyperparameter Selection and Tuning

As we have seen so far, other than the learnable parameters of an NN model, there are plenty of hyperparameters to set and choices to be made, such as the number of layers, the number of neurons for each layer, the activation functions, regularization coefficients, and also training hyperparameters, like learning rate, batch size, number of epochs, and optimizer choice. NN surrogate quality depends critically on these architectural and hyperparameter choices. For this reason, systematic selection methods have become a key part of modern practice.

Fundamentally, the search for optimal values (hyperparameter tuning) can be seen as a surrogate optimization problem: we sample different hyperparameter combinations and monitor the validation loss. Thus, all of the sampling methods discussed in Sec. 1.2 can be valid options. However, in many practical scenarios, performing grid search, random search, or similar techniques is infeasible, as training a single model may already demand significant computational resources. Consequently, one often aims to minimize the number of hyperparameter trials. Bayesian Optimization can be an effective alternative: it models the validation loss as a function of the hyperparameters and uses expected improvement to select promising configurations, typically requiring far fewer evaluations than grid or random search.

In general, it is advisable to start with shallow architectures with moderate widths, then expand depth/width if validation performance plateaus. Let us keep in mind that the universal approximation theorem guarantees that even shallow networks can represent any continuous function; however, deeper networks often achieve better generalization with fewer parameters through hierarchical feature learning. An even better approach, as always, is to check for similar works using similar architectures, to already have a decent, educated guess for good enough hyperparameters.

PART II
Atomistic Modeling

Chapter 3

Theory & Methods for Atomistic Modeling

3.1 Context & Motivation

The landscape of atomistic simulation methods has undergone a noticeable shift in recent years, driven largely by the growing popularity of machine learning interatomic potentials (MLIPs) [7, 59, 9, 10, 11, 60]. As already anticipated in the Introduction, these data-driven models present a solution to the high computational cost of *ab initio* methods. Thanks to their balance between accuracy and speed, MLIPs enable investigations of substantially larger atomic systems and the simulation of phenomena occurring over extended timescales [13].

Scope and Objectives

The central motivation for the present work is the development of an interatomic potential capable of reliably describing pressure-dependent crystal phase transformations, with particular emphasis on accurately capturing kinetic barriers. As we discuss in greater detail in later dedicated sections, a key methodological step we employ is the incorporation of configurations from minimum energy paths linking different metastable states into the MLIP training dataset. This enables robust exploration of transformation mechanisms and nucleation phenomena that would otherwise remain inaccessible to large-scale simulations. While we take germanium (Ge) as an interesting case study due to its potential applications [22], the emphasis lies on establishing and validating an effective development strategy that can be extended to other systems exhibiting complex phase behavior.

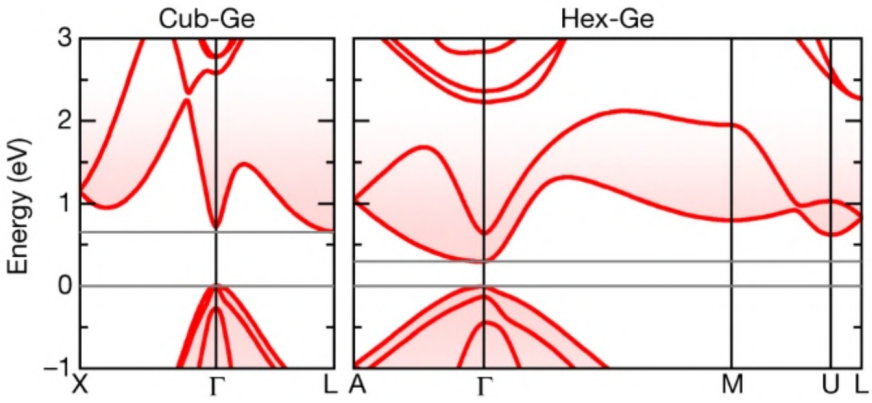

Figure 3.1: Germanium band structure, showing the existence of a direct band gap for the hexagonal diamond phase, rendering it potentially useful for optoelectronic applications. Taken from Ref. [22] with permission.

Germanium as a Model System

Germanium serves as a particularly instructive material system for developing and validating MLIPs designed for phase transition studies, owing both to its rich polymorphic behavior and its technological significance. At ambient conditions, Ge is a semiconductor exhibiting a cubic diamond (CD) structure (space group $Fd\bar{3}m$). Under non-hydrostatic pressure of approximately 10 GPa, as demonstrated through diamond anvil cell experiments and nanoindentation studies, the CD phase transforms to a metallic β-Sn structure ($I4_1/amd$) [61, 62, 63]. This high-pressure phase exhibits remarkable stability across a broad pressure range and, upon further compression, undergoes subsequent transitions to even denser metallic configurations, including simple hexagonal (SH) and hexagonal close-packed (HCP) arrangements [64, 65].

The decompression pathway from the β-Sn phase does not restore the original CD structure; instead, a variety of metastable allotropes emerge, including BC8 ($Ia\bar{3}$), ST12 ($P4_3\bar{2}_1\bar{2}$), and R8 ($R\bar{3}$) phases. Experimental observations suggest that the formation of these specific allotropes depends on the prevailing stress conditions and the rate at which decompression from β-Sn is performed [66, 67, 68, 69, 70]. The ST12 phase has attracted sustained attention due to its distinctive optical properties [71], with recent advances enabling the synthesis of large-scale pure bulk samples [72], the fabrication of ST12 nanoparticles [73, 74], and the production of nanowires [75]. Meanwhile, both BC8 and R8 phases have been observed to transform into a hexagonal diamond (HD) structure

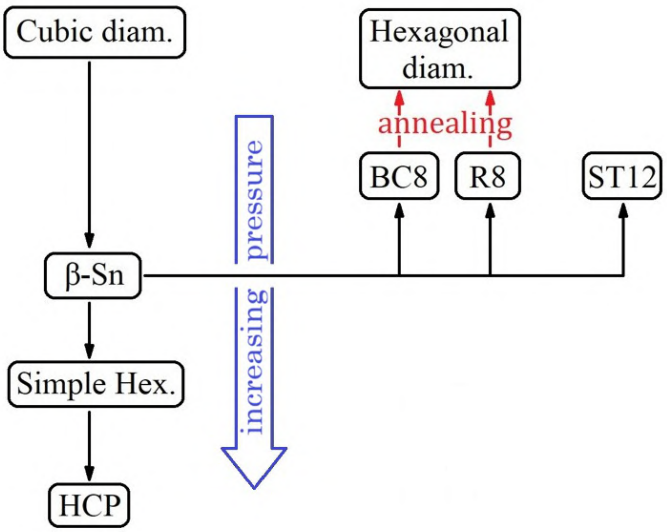

Figure 3.2: Sketched summary of the most relevant crystal phase transitions in Ge, notably the pathway leading from cubic diamond to hexagonal diamond.

under ambient pressure and temperature [76, 77]. This transformation pathway holds considerable technological interest, as HD Ge is predicted to possess a narrow direct bandgap, shown in Fig. 3.1, making it an attractive candidate for infrared optoelectronic applications with potential integration into silicon-based technologies [22]. A summary of relevant crystal phase transitions is reported in Fig. 3.2

Additionally, amorphous germanium (a-Ge) has been extensively studied across various experimental investigations. It can appear as a byproduct during the decompression process from β-Sn, emerges as an intermediate state during pressure-induced transformation to β-Sn, and has been observed undergoing transitions between low-density and high-density amorphous forms [78, 79, 80].

3.2 Density Functional Theory

The first step in the development of an MLIP is preparing a suitable dataset of atomic configurations labeled with energies and eventually forces and virials. The dataset used to train the MLIP is derived from density functional theory (DFT) calculations. In this section, we briefly review the key concepts of DFT, both to introduce the method to readers

unfamiliar with it and to provide a quick refresher for those who are already acquainted with it.

DFT, for roughly half a century, has been the dominant *ab initio* approach in computational quantum mechanics, enabling the study of electronic properties in atoms, molecules, and solids from first principles. The theory reformulates the task of solving the many-electron Schrödinger equation by using the electronic density $n(\mathbf{r})$ as the fundamental variable, and drastically reduces computational complexity for large systems.

The Hohenberg-Kohn-Sham Approach

In quantum mechanics, the starting point for describing any system is the well-known Schrödinger equation, whose solution provides the wavefunction depending on all nuclear and electronic coordinates. The Born–Oppenheimer approximation [81] is typically employed to separate electronic and nuclear motion, thanks to their large mass difference, to reduce the full problem into simpler components.

In the 1960s, Hohenberg, Kohn, and Sham introduced Density Functional Theory (DFT), which hugely simplifies the description of many-electron systems [82, 83]. DFT replaces the interacting electron system with an auxiliary system of non-interacting electrons moving in an effective potential. In this framework, the ground-state electron density of this fictitious non-interacting system is identical to that of the true interacting system under the external potential generated by the nuclei, $v_{\text{ext}}(\mathbf{r})$. This avoids the need to compute the many-body wavefunction explicitly, allowing the energy to be expressed instead in terms of the electron density:

$$E[n] = T_s[n] + \int v_{\text{ext}}(\mathbf{r}) n(\mathbf{r}) d\mathbf{r} + E_{\text{H}}[n] + E_{\text{xc}}[n] \tag{3.1}$$

where $T_s[n]$ is the kinetic energy of the system, $E_{\text{H}}[n]$ is the classical Hartree energy, $E_{\text{xc}}[n]$ is the exchange-correlation term.

Exchange-Correlation Approximations

In practice, the accuracy of the results is limited by the approximations in the exchange-correlation functional $E_{\text{xc}}[n]$ of which the exact form is unknown. Many different functionals have been developed, with the first two basic approaches being local density approximation (LDA) [83] and generalized gradient approximation (GGA) [84, 85], followed by meta-GGA and various hybrid functionals [86, 87, 88, 89]. Still,

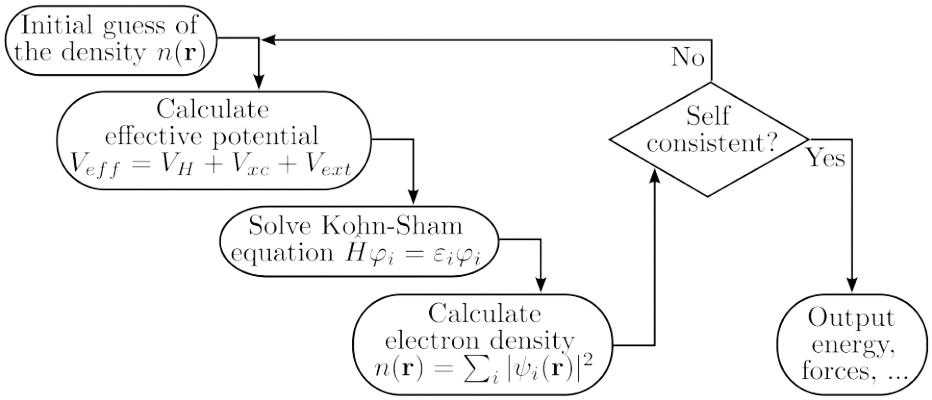

Figure 3.3: Sketch of the iterative self-consistent procedure employed to solve the KS equation. Initially, a guessed electronic density is used to evaluate the effective potential and then to solve the KS equation. The orbitals obtained from the previous step are used to compute a new density and subsequently a new effective potential. The new effective potential is plugged again into the KS problem, and the procedure is iteratively repeated until convergence is reached.

there is no single universally optimal functional as different materials and use cases often favor different choices. Some functionals provide better accuracy for predicting formation energies across crystal structures, while others perform better for estimating bond lengths, and so on.

Self-Consistent Field Procedure

Once the choice of a suitable exchange-correlation functional is made, DFT solutions are found by iteratively updating the density:

$$\sum_{i}^{\text{occ}} |\psi_i^{(k)}(\mathbf{r})|^2 \rightarrow n^{(k+1)}(\mathbf{r}). \tag{3.2}$$

The procedure is repeated until $n^{(k+1)}(\mathbf{r}) \approx n^{(k)}(\mathbf{r})$ within a chosen tolerance. A simplified sketch of the iterative procedure to compute DFT energies and forces of any given atomistic configuration is shown in Fig. 3.3.

Projector Augmented Wave Method

For periodic solids, Kohn-Sham orbitals are expanded in plane waves with energies below a certain cutoff energy (for numerical reasons), and

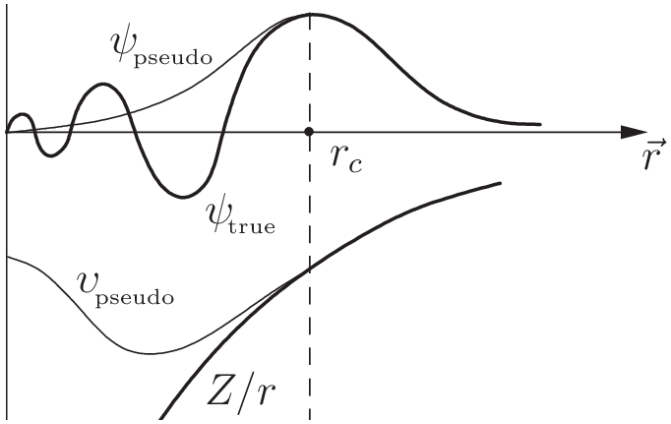

Figure 3.4: Comparison between a long-range $1/r$ potential and a smoother pseudopotential and their corresponding wavefunction.

so-called pseudopotentials are used to efficiently describe interactions between valence electrons and ion cores, reducing the need to explicitly treat tightly bound core states [90]. A modern, commonly used, pseudopotential formulation is the projector augmented wave (PAW) method, which overcomes the plane-wave method's difficulties with highly oscillatory all-electron wavefunctions near nuclei.

Brillouin Zone Sampling

Quantities in periodic crystals, like the electronic density and total energy, require integration over the Brillouin zone (BZ). Numerically, this is performed via summation over a discrete mesh of **k**-points. After carefully choosing the number N_k and distribution of **k**-points, the density at each point and the total energy are sampled as:

$$n(\mathbf{r}) = \frac{1}{N_k} \sum_{\mathbf{k},i} f_{i\mathbf{k}} |\psi_{i\mathbf{k}}(\mathbf{r})|^2 \tag{3.3}$$

where $f_{i\mathbf{k}}$ is the occupation number of band i at **k**. The convergence of computed properties with respect to the **k**-point mesh is essential for reliable results. In this work, the Monkhorst-Pack method [91] was used to generate an optimal k-point mesh for integrations over the BZ.

For a more detailed discussion of the theoretical foundations and practical implementation of density functional theory, the reader is referred to the comprehensive treatments in Density Functional Theory: A Practical Introduction [92] and in Density Functional Theory: An Advanced Course [93].

3.3 Atomistic Simulations and Sampling

MLIPs, like all data-driven models, rely critically on the quality and diversity of their training data. Sampling plays a crucial role in constructing an effective surrogate model, and in this section, we review standard and novel sampling strategies for building datasets suitable for training MLIPs.

Molecular Dynamics Sampling

Molecular dynamics (MD) is a fundamental tool in atomistic modeling, enabling the computational study of particle interactions and their time evolution at the nanoscale. In principle, MD can also be used to generate large datasets: by extracting snapshots from trajectories at a chosen rate, one can obtain numerous configurations for training a model. Moreover, simulations can be conducted under different ensembles (NVT or NPT) and varying temperature, pressure, or volume (using Nose-Hoover thermostats and barostats), producing a diverse set of configurations for more robust training or more targeted conditions.

Despite its versatility, standard MD faces significant limitations. Each MD step must accurately resolve atomic vibrations, typically on the femtosecond timescale. Consequently, even long simulations usually span only nanoseconds to microseconds, for simple systems or highly optimized codes. Many real processes, such as atomic diffusion, phase transformations, and chemical reactions, occur over much longer timescales (seconds, minutes, or more) and are separated by substantial energy barriers. Rare events occur far too infrequently to be captured directly in standard MD simulations, limiting both the physical realism and statistical reliability of the results. To properly sample rare events, different approaches have to be employed.

Enhanced Sampling Methods

A substantial body of work has focused on accelerating the sampling of rare events during dynamics simulations. These efforts have led to the development of a wide range of enhanced sampling techniques [94, 95, 96]. Although these methods are not employed in the present work, we briefly review them to inform interested readers of their existence and general scope.

A common class of approaches introduces a bias potential to enhance exploration of low-probability regions in configuration space [97, 98, 99,

100, 101, 102]. The classic examples, umbrella sampling [103] and metadynamics [100, 104], modify the energy landscape to facilitate sampling and the computation of thermodynamic quantities. Umbrella sampling applies static bias along a reaction coordinate or collective variable (CV), which act as a reduced descriptor of slow modes or essential pathways. The trajectory is forced to sample across energy barriers, and reweighting schemes are applied to recover unbiased statistical distributions. The static nature of the bias offers controlled sampling, but demands careful planning and prior knowledge of transition coordinates. In metadynamics, instead, time-dependent biases are introduced through the addition of Gaussians to the CV space, dynamically filling wells and flattening barriers to encourage transitions. Metadynamics also enables automated exploration of multidimensional landscapes.

More recently, new enhanced sampling frameworks have been introduced. The on-the-fly probability enhanced sampling (OPES) method [105], and the OPES-flooding variant [106], represent a significant improvement over metadynamics, offering faster convergence and easier usage. However, these methods rely critically on well-chosen CVs, whose design can be far from straightforward. For instance, in systems with multiple complex reaction pathways, selecting suitable CVs becomes particularly challenging [95, 107], and convergence may be slow if the chosen CVs fail to capture the relevant mechanisms.

A second major family of techniques avoids the use of external bias altogether. They instead aim to reconstruct transition pathways and kinetics by partitioning the high-dimensional landscape into interfaces and running many short, independent simulations [108, 109, 110, 111, 112]. Transition path sampling (TPS) [113], and its variants [114], generate unbiased reactive trajectories via Markov-chain Monte Carlo in trajectory space to produce accurate transition path ensembles. Despite their conceptual elegance, these methods can suffer from high computational cost and limited sampling efficiency, which may restrict their practicality for complex systems [115].

Beyond classical rare-event frameworks, machine-learning–based approaches are advancing rapidly. NN–assisted enhanced sampling has demonstrated significant potential [116, 117]. Other developments include FlowRES [115], a physics-informed framework that employs unsupervised normalizing flow networks to boost Monte Carlo efficiency by generating high-quality, non-local proposal moves for rare-event sampling.

A final mention is about diffusion models: originally developed for

generative modeling of images [118, 119, 120, 121, 122] have recently been extended to molecular systems. In a recent work [123], a diffusion-based generative model for enhanced sampling of rare events has been proposed, capable of generating entire dynamical trajectories of molecules' dynamics, enhancing sampling of rare events.

Saddle-Point-Search Algorithms for Rare Events

To directly identify transition states and the associated activation barriers accurately, saddle-point search algorithms have been developed. Accurately locating saddle points on potential energy surfaces is crucial because, according to transition state theory, reaction rates depend exponentially on the free-energy barrier, making even small errors in barrier estimation lead to orders-of-magnitude discrepancies in predicted dynamics. The characterization of rare event transitions has been a focus of a lot of research in the past [124, 125, 126].

Two widely used algorithms in this category, compared in Fig. 3.5, are the nudged elastic band (NEB) method [127, 128] and the dimer method [129]. In the NEB approach, the initial and final minima must be known, and they are connected by a set of intermediate "images" that form an initial guess for the reaction pathway. These images are linked with virtual springs and iteratively relaxed to reveal the minimum energy path, while a climbing-image refinement step locates the saddle point and improves the accuracy of the barrier estimate. However, NEB has important limitations: both endpoint minima must be specified in advance, and the identified transition state depends on the initially guessed pathway, which may cause certain mechanisms to be missed, especially those associated with the lowest barriers, which typically dominate the kinetics.

The dimer method helps overcome these limitations [131]. Unlike NEB, it requires no predefined final state. Instead of mapping a full path, the dimer method rotates a pair of configurations within the local basin of a given minimum energy configuration until the direction of the unstable mode is found, then climbs along the path to the saddle point. Because it requires only force calculations and minimal prior knowledge, the dimer method is highly efficient for the direct search of transition states even in multidimensional systems. By repeatedly initializing from the same basin, different escape pathways can be explored, allowing systematic identification of the lowest-energy barrier and associated transition mechanism.

These and other similar methods [132] provide a direct way to get

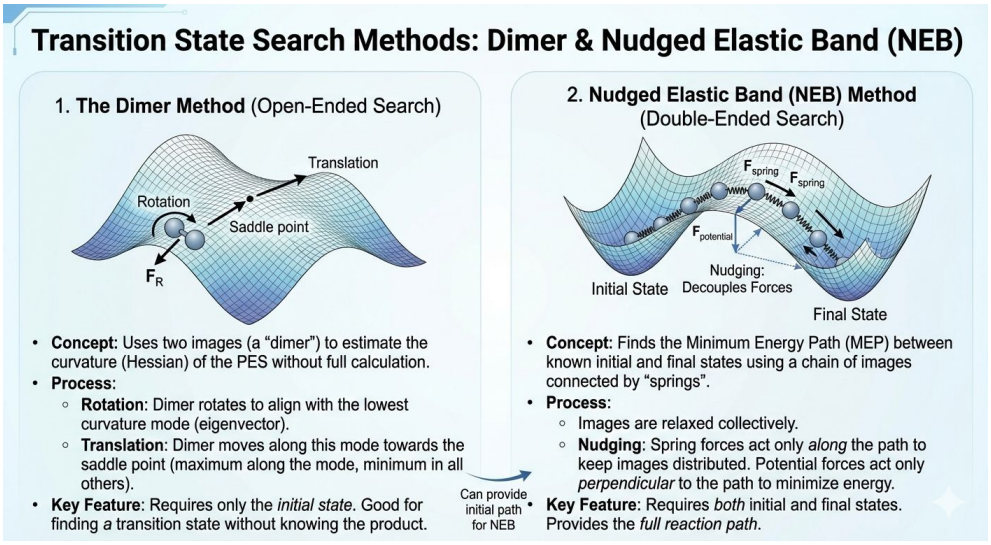

Figure 3.5: Sketch and comparison of the nudged elastic band and dimer methods. Taken from Ref. [130] with permission.

activation energies and understand transition mechanisms at the atomic level, complementing molecular dynamics simulations.

3.4 Solid-state saddle-point search approaches

A major challenge in extending the NEB and dimer methods to variable-cell systems lies in achieving a consistent coupling between atomic and lattice degrees of freedom. To address this issue, Refs. [133, 134] propose a unified formulation in which atomic positions $\mathbf{r}$ and the cell matrix $\mathbf{h}$ are treated within a single generalized coordinate space. This formulation leads to the generalized solid-state NEB method [133] and the solid-state dimer method [134], which we extensively employ to investigate solid–solid crystalline phase transitions.

Atomic and lattice variables coupling

Variations of the cell are naturally expressed in terms of strain variables, which differ in dimensionality from atomic displacements. To combine these quantities into a common displacement vector, a Jacobian (or metric) is introduced, leading to the definition

$$\Delta\mathbf{R} = \{J\varepsilon, \Delta\mathbf{r}\}. \tag{3.4}$$

Here, J denotes the Jacobian, $\varepsilon = \mathbf{h}^{-1}\Delta\mathbf{h}$ is the strain tensor, and $\Delta\mathbf{r}$ represents the change in atomic positions expressed in Cartesian coordinates. The strain is evaluated relative to the reference (undeformed) cell. The cell matrix $\mathbf{h}$ and the strain tensor ε are 3×3 matrices, while the atomic displacements $\Delta\mathbf{r}$ form an $N \times 3$ matrix, with N denoting the number of atoms in the supercell. Accordingly, vectors in the generalized space $\Delta\mathbf{R}$ may be viewed as $(N + 3) \times 3$ matrices. For the purposes of the solid-state dimer method, however, these vectors are flattened when computing norms and dot products.

To ensure a proper separation between atomic and lattice contributions, atomic displacements are defined in relative coordinates with respect to the cell vectors. With this choice, a pure strain ε in the generalized space does not induce any spurious atomic motion. This separation is achieved by expressing $\Delta\mathbf{r}$ as fractional displacements along the lattice vectors and subsequently converting them to Cartesian coordinates, using $\mathbf{h}$ as the reference cell, consistent with the strain definition.

A crucial aspect of constructing the generalized coordinate space is the selection of an appropriate Jacobian. The role of J is to provide a metric that balances the contributions of strain and atomic motion within a single displacement vector. The guiding principle adopted here is that distances in the generalized space should be independent of the overall size and shape of the supercell. Following Refs. [133, 134], the Jacobian is chosen as

$$J = L\sqrt{N}, \qquad L = \left(\frac{\Omega}{N}\right)^{1/3}, \tag{3.5}$$

where L represents the average interatomic spacing derived from the mean atomic volume, and Ω is the supercell volume. The Jacobian has dimensions of length, compensating for the dimensionless nature of the strain tensor. Its value is determined during initialization and remains fixed throughout the calculation. A constant Jacobian is sufficient, since the precise numerical value is less important than its scaling with system size. Moreover, saddle points are stationary and therefore invariant under different choices of the Jacobian.

Within this framework, the distance in generalized configuration space is defined as

$$||\Delta\mathbf{R}|| = \sqrt{NL^2||\varepsilon||^2 + ||\Delta\mathbf{r}||^2}. \tag{3.6}$$

This expression illustrates how the Jacobian sets the relative weighting between lattice strain and atomic displacements. For instance, when

a unit cell is replicated to form a larger supercell while preserving identical atomic motion within each unit, the strain contribution increases linearly with N due to the $\sqrt{N}$ dependence of J (Eq. 3.5). Similarly, the contribution from atomic displacements also scales linearly with the number of atoms. As a result, the ratio between the two terms remains unchanged, ensuring a consistent representation of atomic and lattice degrees of freedom irrespective of supercell size. It can be straightforwardly shown that dot products and angles between vectors in the generalized space are likewise invariant under supercell replication, confirming the absence of metric distortion.

Taking the derivative of the energy with respect to the generalized coordinates $\mathbf{R}$ yields the corresponding generalized force,

$$\mathbf{F} = \left\{ \frac{\Omega}{J} \sigma^{\text{cauchy}}, \mathbf{f} \right\}, \tag{3.7}$$

where $\mathbf{f}$ denotes the atomic forces and σ^{cauchy} is the Cauchy stress tensor.

Solid-state NEB method

In the conventional NEB formulation, the force acting on image i is given by the sum of the true force component perpendicular to the path and an elastic spring force acting along the path, $\mathbf{F}_i^{S_{||}}$, defined as

$$\mathbf{F}_i^{S_{||}} = k(|\Delta\mathbf{r}_-| - |\Delta\mathbf{r}_+|)\hat{\tau}_i, \tag{3.8}$$

where k denotes the spring constant and $\hat{\tau}_i$ is the upwind tangent, chosen based on the neighboring image with higher potential energy. The displacement vectors connecting image i to its neighbors are

$$\begin{aligned} \Delta\mathbf{r}_+ &= \mathbf{r}_{i+1} - \mathbf{r}_i, \\ \Delta\mathbf{r}_- &= \mathbf{r}_i - \mathbf{r}_{i-1}. \end{aligned} \tag{3.9}$$

When extending NEB to the solid-state case, lattice degrees of freedom must be included explicitly. In this context, special care is required to properly decouple atomic displacements from cell deformations when evaluating $\Delta\mathbf{R}_+$, $\Delta\mathbf{R}_-$, and the tangent direction using Eq. 3.4.

If atomic displacements $\Delta\mathbf{r}$ are computed directly in Cartesian coordinates, changes in the cell geometry would spuriously contribute to atomic motion due to the periodic representation of the atoms. To avoid this coupling, atomic positions are instead treated in direct (fractional) coordinates, defined with respect to the lattice vectors $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$.

Displacement vectors are first computed in this fractional representation and subsequently transformed back to Cartesian coordinates for force projection.

Since the configurations at images i and $i+1$ generally correspond to different cell geometries, the conversion between direct and Cartesian coordinates is performed using an average cell. Specifically, atomic positions $\mathbf{r}_i$ and $\mathbf{r}_{i+1}$ are first expressed in direct coordinates using their respective cell matrices $\mathbf{h}_i$ and $\mathbf{h}_{i+1}$. The fractional displacement $\Delta\mathbf{r}_+^{\text{direct}}$ is then obtained by subtraction and converted back to Cartesian coordinates using the averaged cell matrix

$$\bar{\mathbf{h}}_+ = \frac{1}{2}(\mathbf{h}_i + \mathbf{h}_{i+1}). \tag{3.10}$$

In the absence of lattice deformations, this procedure reduces to the standard NEB definition of distances. A similar ambiguity arises in defining the strain between two images, as the strain depends on the choice of reference cell. For consistency, the vector connecting image i to image $i+1$ must be the negative of that connecting $i+1$ to i. However, using the straightforward definition

$$\varepsilon = \mathbf{h}^{-1} \cdot (\mathbf{h}^{\text{def}} - \mathbf{h}), \tag{3.11}$$

with $\mathbf{h}^{\text{def}}$ denoting the deformed cell, leads to

$$\mathbf{h}_i^{-1} \cdot (\mathbf{h}_{i+1} - \mathbf{h}_i) \neq -\mathbf{h}_{i+1}^{-1} \cdot (\mathbf{h}_i - \mathbf{h}_{i+1}). \tag{3.12}$$

To enforce antisymmetry between adjacent images, the strain is therefore defined using an averaged inverse cell matrix,

$$\bar{\varepsilon}_+ = \frac{1}{2}(\mathbf{h}_{i+1}^{-1} + \mathbf{h}_i^{-1}) \cdot (\mathbf{h}_{i+1} - \mathbf{h}_i). \tag{3.13}$$

Following the same rationale, the Jacobian J is kept fixed throughout the NEB optimization. Accordingly, in Eq. 3.5 the cell volume entering the definition of J is taken as the average between the initial and final states,

$$\bar{\Omega} = \frac{1}{2}(\Omega_{\text{initial}} + \Omega_{\text{final}}). \tag{3.14}$$

Solid-state dimer method

A dimer configuration is specified by a central point $\mathbf{R}$ and an orientation vector τ. Within the generalized space combining lattice and

atomic degrees of freedom, $\mathbf{R}$ consists of the cell matrix $\mathbf{h}$ and the atomic coordinates $\mathbf{r}$. Consistent with Eq. 3.4, the orientation vector is written as $\tau = \{\tau_\varepsilon, \tau_\mathbf{r}\}$, where τ_ε and $\tau_\mathbf{r}$ represent the strain and atomic components, respectively. Alternatively, the dimer may be described by two images of the system, $\mathbf{R}_1$ and $\mathbf{R}_2$, positioned symmetrically about $\mathbf{R}$ and separated along τ by a small distance $||\Delta\mathbf{R}||$.

The first endpoint $\mathbf{R}_1$ is constructed from $\mathbf{R}$ and τ as follows. The cell matrix at the endpoint, $\mathbf{h}_1$, is obtained by first defining the associated strain,

$$\varepsilon = \frac{||\Delta\mathbf{R}||}{J}\tau_\varepsilon, \qquad \mathbf{h}_1 = \mathbf{h}\varepsilon + \mathbf{h}. \tag{3.15}$$

Application of this strain rescales the Cartesian atomic positions from $\mathbf{r}$ to a new set of coordinates, denoted $\mathbf{r}'$. Throughout this transformation, atomic positions expressed in fractional coordinates relative to the lattice vectors remain unchanged. Once the new cell geometry is established, the atomic positions are further displaced along the atomic component of the dimer direction,

$$\mathbf{r}_1 = \mathbf{r}' + ||\Delta\mathbf{R}||\tau_r. \tag{3.16}$$

With the generalized definitions of coordinates, displacements, and forces in place, the remainder of the solid-state dimer algorithm proceeds analogously to the standard dimer method. The dimer orientation τ is iteratively rotated based on the force difference between the two endpoints; after each rotation, the endpoint positions and corresponding forces are updated. Once τ converges to the direction of minimum curvature, the dimer center is advanced uphill along τ and downhill in all orthogonal directions using a force-based optimization scheme, until convergence to a saddle point is achieved.

3.5 Computational Framework and Implementation Details

Having briefly reviewed some key aspects of density functional theory and having briefly summarized different simulation and sampling approaches, we now describe the practical implementation adopted in our work. In particular, the selection of appropriate computational parameters, done after convergence criteria checks, plays a critical role in ensuring that the machine learning potential training is based on reliable reference data.

This reflects the usual saying *"garbage in, garbage out"*, highlighting that an ML model's predictions can only be as reliable as the reference data on which it is trained.

DFT using VASP

All first-principles electronic structure calculations were performed using the *Vienna Ab Initio Simulation Package* (VASP). Within this framework, the PAW method was employed to describe the interaction between ionic cores and valence electrons, with the Ge $3d\,4s\,4p$ electrons explicitly treated as valence states. The exchange-correlation functional adopted throughout this work is the Perdew–Burke–Ernzerhof (PBE) formulation [135], selected as it is a popular choice given its reasonable accuracy over a wide range of systems.

Systematic convergence tests were conducted to establish an appropriate balance between computational accuracy and practical efficiency. Based on these assessments, the plane wave basis set truncation energy was established at 500 eV, while the Brillouin zone sampling employed a uniform k-point grid spacing of $0.14\,\text{Å}^{-1}$. Formation energies for each crystalline phase were determined by computing the per-atom energy difference relative to the cubic diamond structure, which serves as the thermodynamic reference state at ambient conditions.

Convergence analysis using the adopted k-point grid spacing revealed that formation energies for the semiconducting phases, including 4H-HD, 2H-HD, BC8, ST12, and R8, reached convergence within 1 meV/atom upon further refinement of computational parameters. In contrast, denser metallic structures such as β-Sn, Fmmm, SH, HCP, BCC, and FCC exhibited residual oscillations in computed energies, with deviations reaching approximately 5 meV/atom in certain instances. This level of precision was deemed acceptable, particularly considering that substantially larger discrepancies in formation energies would arise from the choice of alternative exchange-correlation functionals. Given the central focus on pressure-dependent transformations, convergence with respect to predicted pressures was also monitored. For semiconducting phases, pressure convergence reached approximately 0.01 GPa, while metallic structures, more demanding in their convergence behavior, displayed oscillations approaching ≈ 1 GPa in isolated cases.

Molecular dynamics, optimizations, and saddle-point searches

Molecular dynamics trajectories were generated using the *Large-scale Atomic/Molecular Massively Parallel Simulator* (LAMMPS) [136], interfaced with the DeePMD [25] extension to enable evaluation of energies, forces, and stresses through the trained machine learning potential. Configuration sampling was performed within both the canonical (NVT) and isothermal-isobaric (NPT) statistical ensembles, employing an integration timestep of 1 fs. Temperature and pressure control were implemented via Nosé–Hoover thermostat and barostat algorithms, configured with relaxation timescales of 0.5 ps and 5.0 ps, respectively.

For structural relaxations at the atomic scale, the Fast Inertial Relaxation Engine (FIRE) [137] optimization algorithm was utilized within the Atomic Simulation Environment (ASE) [138]. In these calculations, the trained MLIP served as the computational engine for evaluating potential energies, interatomic forces, and stress tensors, enabling efficient geometry optimization of both bulk and defective configurations.

The generalized solid-state NEB method [133] incorporating the climbing-image refinement was adopted for identifying minimum energy pathways between crystal phase transitions involving both atomic and cell. The implementation employed in this work utilizes the TSASE code [139], which provides an interface to the ASE computational framework. Additional transition states were also investigated using the generalized solid-state dimer method [134]. Aiming at computing activation energies for solid-state phase transitions at different desired stress conditions, both the solid-state NEB and dimer methods were formulated in an enthalpy landscape by incorporating an external stress contribution. We achieve this by modifying Eq. 3.7 as:

$$\mathbf{F} = \left\{ \frac{\Omega}{J}(\sigma^{\text{cauchy}} + \sigma^{\text{external}}), \mathbf{f} \right\}. \tag{3.17}$$

Chapter 4

Overview of Machine Learning Interatomic Potentials

4.1 Brief history of the main architectures

Behler–Parrinello Neural Network Potential (NNP)

The development of machine learning interatomic potentials originates from the recognition that traditional empirical potentials, typically defined by a limited number of parameters, suffer from restricted flexibility and limited accuracy. Following early efforts in the 1990s, a major breakthrough occurred in 2007 with the seminal work of Behler and Parrinello [7]. In their proposed framework, the total energy of a system is decomposed into contributions from individual atoms, each determined solely by its local atomic environment within a finite cutoff radius.

This idea leads to a permutation-invariant formulation of the total energy,

$$E = \sum_i E_i, \tag{4.1}$$

where E_i denotes the energy associated with atom i.

More specifically, for a given atomic configuration, a descriptor is constructed for each atom to encode the geometric arrangement of its neighboring atoms. These descriptors have to be designed to be invariant under translations, rotations, and permutations of identical atoms, and to vary smoothly with atomic positions. Behler and Parrinello introduced the Atomic-Centered Symmetry Functions (ACSFs) to fulfill these requirements. ACSFs capture the influence of neighboring atoms within

a cutoff radius through a combination of radial and angular symmetry functions, defined as

$$G_i^{\mathrm{rad}} = \sum_{j \neq i}^{N_{\mathrm{atom}} \in R_c} e^{-\eta (R_{ij} - R_s)^2} \, f_c(R_{ij}), \tag{4.2}$$

$$G_i^{\mathrm{ang}} = 2^{1-\zeta} \sum_{j,k \neq i \, j<k}^{N_{\mathrm{atom}} \in R_c} (1 + \lambda \cos \theta_{ijk})^{\zeta} \; e^{-\eta (R_{ij}^2 + R_{ik}^2 + R_{jk}^2)} \, f_c(R_{ij}) f_c(R_{ik}) f_c(R_{jk}), \tag{4.3}$$

with the cutoff function

$$f_c(R_{ij}) = \begin{cases} \frac{1}{2} \left[\cos\left(\pi \frac{R_{ij}}{R_c} \right) + 1 \right], & R_{ij} < R_c, \\ 0, & R_{ij} \geq R_c. \end{cases} \tag{4.4}$$

Here, i, j, and k label atoms, while η, R_s, λ, and ζ are tunable hyperparameters controlling the resolution of the local environment representation. The resulting symmetry function values are then provided as inputs to a feed-forward neural network, which outputs an atomic energy contribution for each atom in the system. The sum of these contributions yields the total potential energy, completing the NNP framework.

Gaussian Approximation Potential (GAP)

Another prominent class of MLIPs is based on Gaussian process regression (GPR). As in Behler–Parrinello–type approaches, GPR-based potentials rely on the construction of descriptors that encode the local atomic environment around each atom. The Gaussian Approximation Potential (GAP), originally introduced in 2010 by Bartók *et al.* [41], combines GPR with bispectrum-based descriptors to model interatomic interactions.

To overcome limitations in accuracy and transferability observed in the original formulation, the same authors in 2013 introduced the Smooth Overlap of Atomic Positions (SOAP) descriptor [140]. In contrast to earlier representations that explicitly encode local atomic densities, SOAP defines a kernel that directly measures the similarity between pairs of local atomic environments. This formulation is particularly well-suited for kernel-based regression methods and has become a standard descriptor in modern GPR-based MLIPs.

A key advantage of GPR-based approaches is their Bayesian nature, which provides an intrinsic estimate of prediction uncertainty. This feature enables active and on-the-fly learning strategies, whereby configurations associated with large uncertainties or errors are selectively added to

the training set after validation via first-principles calculations. The effectiveness of this framework has been demonstrated by the development of highly accurate and transferable models, most notably the general-purpose interatomic potential for silicon reported in 2018 [10].

Linear expansion of basis functions

An alternative to NN-based and kernel-based MLIPs is provided by approaches that construct explicit, atom-centered descriptors and express the potential energy surface as a linear combination of generalized basis functions. Compared to kernel methods, these models typically offer improved computational scaling with system size and, in many cases, enhanced transferability and generalization.

It has been shown that many widely used atomic descriptors, including SOAP, can be formally derived as special cases of a unified polynomial expansion of the atomic neighbor density known as the Atomic Cluster Expansion (ACE) [141]. Using this descriptor framework fall several prominent linear models, such as the Spectral Neighbor Analysis Potential (SNAP) [142] introduced in 2014, and the Moment Tensor Potentials (MTP) [143] developed in 2016.

In the SNAP formalism, the local environment of each atom is represented through bispectrum components obtained by projecting the neighbor density onto a basis of four-dimensional hyperspherical harmonics. These bispectrum components coincide with the bond-orientational order parameters originally employed in the GAP framework. However, in contrast to GAP, SNAP assumes a linear dependence of the atomic energy on these descriptors. The corresponding linear coefficients are determined via weighted least-squares regression against DFT reference data, enabling robust and automated fitting to large datasets.

This linear formulation significantly reduces computational cost relative to kernel-based approaches, effectively positioning SNAP as a bridge between traditional analytic interatomic potentials and high-accuracy machine-learning models. The resulting efficiency has enabled simulations at unprecedented scales, exemplified by molecular dynamics studies involving up to one billion carbon atoms undergoing pressure-induced amorphous-to-BC8 phase transitions [144].

4.2 The DeePMD framework

For the development of the MLIP used to investigate the pressure-induced phase transitions of germanium, we employed DeePMD-kit [25, 23, 24] developed in 2018. Like other NN-based potentials [7, 145], DeePMD relies on a feed-forward architecture to fit energies. However, it introduces a key innovation: instead of manually designing and tuning symmetry functions, DeePMD learns them automatically through one or more embedded neural networks. These embedded networks generate descriptor vectors that are constructed to respect all relevant physical symmetries. This approach offers substantial flexibility and eliminates the laborious, time-consuming process of crafting custom symmetry functions. The resulting descriptors are then passed to the fitting network, which performs the final regression of the potential energy surface.

Here, we examine in detail how the end-to-end, symmetry-preserving interatomic potential model used in this work operates. Although the framework provides a complete pipeline, understanding its underlying components is essential: several hyperparameters still require careful, informed tuning to achieve reliable and accurate performance.

Models' pipeline

The deep potential (DP) formalism decomposes the system's total energy into atomic contributions, with each atomic environment represented through a local, geometric descriptor. Specifically, a multilayer neural network is employed to regress the mapping from atomic configurations to energy. For every local atomic subsystem, an embedding network converts the geometric and chemical environment into a descriptor matrix, which the fitting network subsequently maps to a local energy value. From the combined output across all atomic environments, we get the system's predicted total energy.

More in detail, starting from a given atomic configuration with N atoms and $\mathbf{r} = \{r_1, ..., r_N\}$ set of atomic coordinates, we want to obtain the PES based on these raw atomic coordinates. To do this, the first step is to start with building for each atom i its local environment matrix, which includes $N_{r_c}^{(i)}$ neighbors inside a chosen cutoff r_c. These local environments $\mathbf{R}_i$ are the input of the embedded networks, which output the embedding matrix $g_i(\mathbf{R}_i)$. Then, embedding matrices are used to compute the actual descriptors $\{D(g_i(\mathbf{R}_i))\}$ of the local environment. These descriptors in turn are the input of the fitting network $F(\{D(g_i(\mathbf{R}_i))\})$, which provides the atomic energy, and hence the total energy is given by:

$$E = \sum_i F\left(\{D(g_i(\mathbf{R}_i))\}\right) \tag{4.5}$$

The local environment matrix

This local environment matrix is defined as

$$\mathbf{R}_i = \{\tilde{\mathbf{r}}_{1,i}^T, ..., \tilde{\mathbf{r}}_{N_{rc}^{(i)},i}^T\} \quad , \quad \tilde{\mathbf{r}}_{j,i} = \{s(r_{j,i}), \hat{x}_{j,i}, \hat{y}_{j,i}, \hat{z}_{j,i}\} \tag{4.6}$$

where $\mathbf{r}_{j,i} = \mathbf{r}_j - \mathbf{r}_i$, j is the index referring to the j-th neighboring atom, $\hat{x}_{j,i} = s(r_{j,i})\, x_{j,i}/r_{j,i}$ is a rescaled coordinate (and similarly for $\hat{y}$ and $\hat{z}$), and $s(r_{j,i})$ is a smoothing function given by:

$$s(r_{j,i}) = \begin{cases} \frac{1}{r_{j,i}} & \text{for } r_{j,i} < r_{cs} \\ \frac{1}{r_{j,i}} \left\{ \frac{1}{2}(\cos(\pi \frac{r_{j,i} - r_{cs}}{r_c - r_{cs}} + \frac{1}{2}) \right\} & \text{for } r_{cs} < r_{j,i} < r_c \\ 0 & \text{for } r_c < r_{j,i} \end{cases} \tag{4.7}$$

where r_{cs} is another internal smoothing cutoff.

Constructed in this way, the local environment matrix $\mathbf{R}_i$ is of dimension $N_{r_c}^{(i)} \times 4$, so the dimension is different for each atom according to the number of its neighbors. Since the embedding networks have a fixed number of input nodes, the size of the local environment matrices is fixed to all atoms to $N_{max} = \max(N_{r_c}^{(i)})$ in the dataset. By doing so, the extra rows are padded to zero in case of atoms with fewer neighboring atoms.

Following, we want to impose two important symmetries on the local environment matrices: permutation and translation symmetries. $\mathbf{R}_i$ already preserves the translation symmetry as it contains only information about distances between the i-th and neighboring atoms. To impose also permutation symmetry, the neighbors in the local environment matrix are ordered from the nearest to the most distant one.

With the $\mathbf{R}_i$ now defined, the next step is to construct the actual descriptors. This is achieved through two-body and three-body embedding networks, which encode local geometric information at different levels of complexity. We outline these embedding schemes below.

Two-body embedding

The two-body embedding matrix $g_i^2(\mathbf{R}_i)$ of the i-th atom is defined as a $N_m \times M_2$ matrix with j-th row given by:

$$(g_i^2)_j = (G_1(s(r_{j,i})), ..., G_{M_2}(s(r_{j,i}))) \tag{4.8}$$

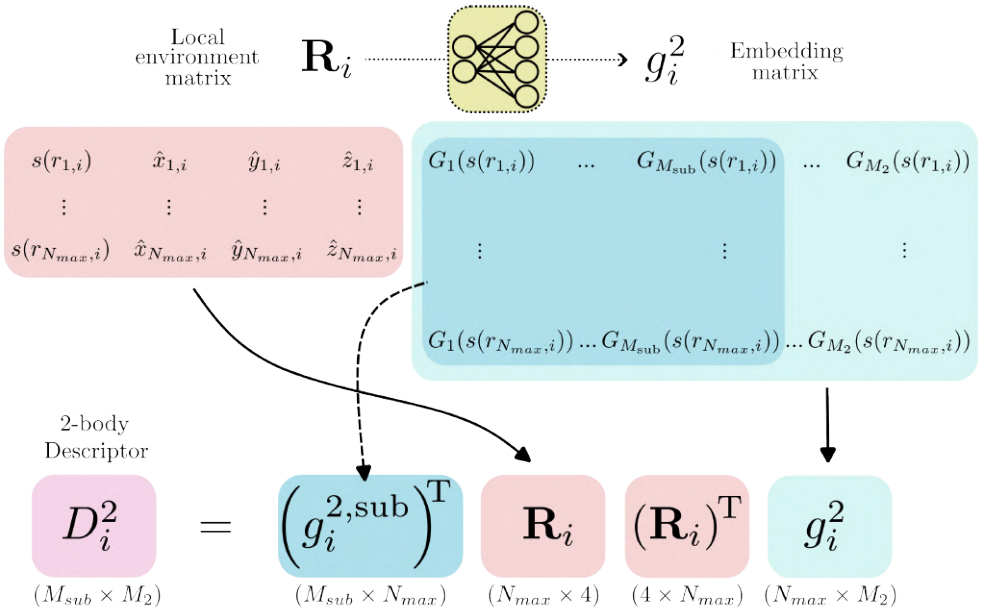

Figure 4.1: Schematic representation of the construction of the two-body descriptor. For each atom i, pairwise distances $r_{j,i}$ are first mapped through the embedding network G to build the two-body embedding matrix g_i^2. The descriptor D_i^2 is then obtained by combining a submatrix of g_i^2 with the neighbor coordinate matrix $\mathbf{R}_i$ and its transpose, ensuring rotational invariance and fixed dimensionality.

where G is the actual embedding network, whose parameters are optimized during the training together with the fitting network. More in detail, G is a fully connected NN acting as a map from the input $s(r_{j,i})$ into M_2 number of outputs.

With this other building block, the actual descriptor is defined as:

$$D_i^2 = \frac{1}{N_{max}^2} g_i^{2,sub} \mathbf{R}_i (\mathbf{R}_i)^{\text{T}} g_i^2 \tag{4.9}$$

where $g_i^{2,sub}$ is a submatrix of g_i^2. For more details on this procedure, you can check Ref. [146], and Fig. 4.1, illustrating a schematic representation of the building process of the two-body descriptors.

Other than the permutation and the translation symmetries discussed earlier, D_i^2 also preserves the rotational symmetry, as it depends only on the two-body embedding matrix and the product of the local environment matrix. Indeed, the former (g_i^2) depends only on the distance between atoms. The latter ($\mathbf{R}_i(\mathbf{R}_i)^{\text{T}}$) is invariant with respect to rotation as it is

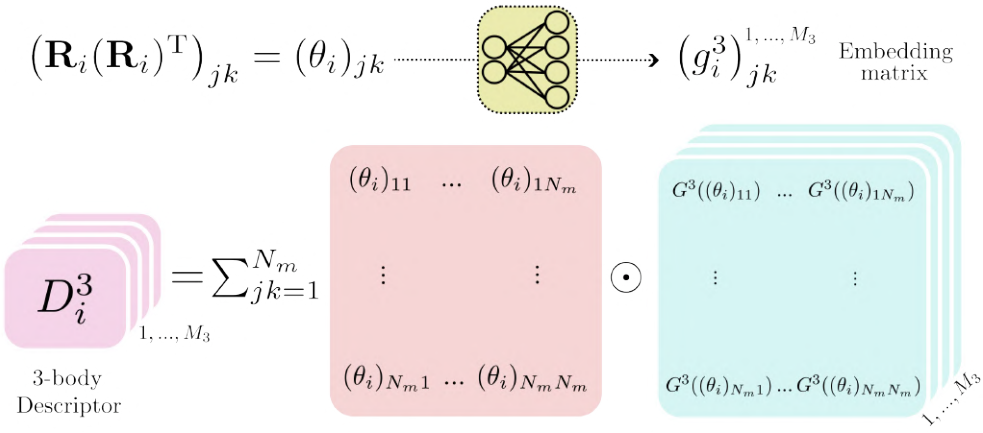

Figure 4.2: Schematic illustration of the construction of the three-body descriptor. Angular information between pairs of neighbors is encoded in the matrix $(\theta_i)_{jk}$, which is mapped through the embedding network G^3 to form the tensor g_i^3. The final descriptor D_i^3 is obtained by contracting $(\theta_i)_{jk}$ with g_i^3, yielding a rotationally invariant three-body representation of the local atomic environment.

known to be a so-called overcomplete base [147].

Three-body embedding

In a similar fashion to the two-body embedding matrix, the three-body embedding matrix is built, but with different input. In this other matrix, the information of the local environment is encoded considering the angle formed by two neighboring atoms. Firstly, we define $(\theta_i)_{jk}$ as the element matrix of the inner product of the local environment matrix with itself:

$$(\theta_i)_{jk} = (R_i(R_i)^{\mathrm{T}})_{jk} = s(r_{j,i})s(r_{k,i})\frac{\mathbf{r}_{j,i}\mathbf{r}_{k,i}}{r_{j,i}r_{k,i}}. \tag{4.10}$$

Then, the three-body embedding element corresponding to the i-th atom is given by:

$$(g_i^3)_{jk} = (G_1^3((\theta_i)_{jk}), ..., G_{M_3}^3((\theta_i)_{jk})) \tag{4.11}$$

where M_3 is the number of outputs of the G^3 embedding network, and whose parameters are optimized during the learning process. Note that $(g_i^3)_{jk}$ is a rank-three tensor.

The actual three-body descriptor D^3 is a function of $(\theta_i)_{jk}$ and $(g_i^3)_{jk}$:

$$D_i^3 = \frac{1}{N_{max}^2} \sum_{j,k=1}^{N_m} (\theta_i)_{jk} (g_i^3)_{jk}. \tag{4.12}$$

A schematic representation of this procedure is shown in Fig. 4.3.

D^3 preserves all the nice physical symmetries of permutations, translations, and rotations, as it has been shown in Ref. [146], based on the same arguments discussed for D^2.

The fitting network

Once the descriptors are constructed, they are concatenated and flattened into a single one-dimensional vector that serves as the input to the fitting network. The fitting network itself is a standard fully connected neural network: its hidden layers use hyperbolic tangent activation functions, while the output layer applies a linear transformation to produce a scalar energy contribution.

The parameters of this network, together with those of the embedding networks, are optimized using stochastic gradient descent with the ADAM algorithm [53]. At each iteration, the model is updated using a randomly shuffled minibatch of configurations drawn from the training set. During the optimization, the weights of the network are updated according to a certain custom loss function, which will be discussed, together with the specific training protocol we used, in Sec. 5.1.

4.3 Recent developments and future directions

Graph Neural Network (GNN) Potentials

In recent years, graph neural networks (GNNs) have emerged as a dominant paradigm for constructing machine-learning interatomic potentials. In these models, atomic systems are represented as graphs, where atoms correspond to nodes and interatomic interactions are encoded through edges. Local atomic environments are learned via message-passing operations, in which information is exchanged between neighboring atoms to construct expressive representations that depend on both chemical identity and local geometry. These representations are subsequently processed by a readout layer to predict atomic energy contributions.

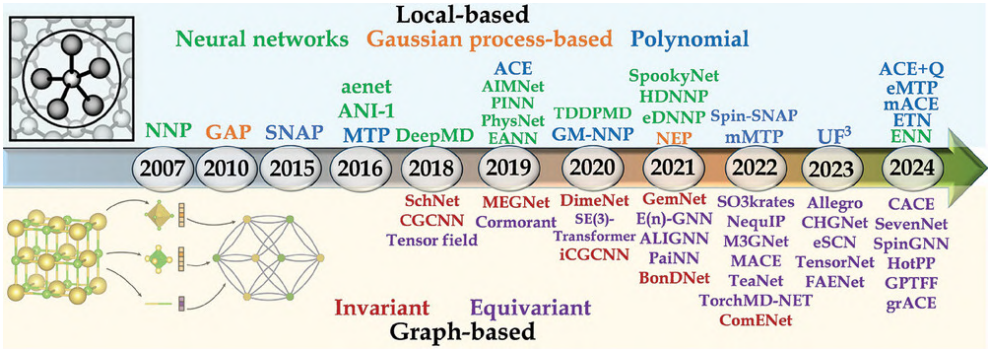

Figure 4.3: Timeline of the development of the main MLIP models. Taken from Ref. [167] with permission.

A defining feature of GNN-based MLIPs is their explicit treatment of physical symmetries. At a minimum, such models enforce translational and rotational invariance of the predicted energy. More advanced architectures also maintain equivariance of intermediate representations and spatial outputs, ensuring that vectorial and tensorial quantities transform correctly under rotations. Although enforcing equivariance generally increases computational cost, it has been shown to substantially improve data efficiency and generalization performance [148, 149].

Early GNN-based potentials focused on distance-based, rotationally invariant formulations. Notable examples include SchNet [150, 151, 152], which learns continuous-filter convolutions from interatomic distances, and the Crystal Graph Convolutional Neural Network (CGCNN) [153], designed specifically for periodic crystal structures.

Subsequent developments introduced directional and angular information to achieve equivariance. One approach, proposed by Satorras *et al.* [154], updates edge features equivariantly during message passing. Alternative strategies compute angular features explicitly, as in DimeNet [155] and its successor GemNet [156, 157], enabling accurate modeling of directional bonding effects.

A further class of models achieves full E(3)-equivariance by leveraging steerable three-dimensional convolutions and Clebsch–Gordan tensor products [158, 159, 160]. Prominent examples include NequIP [148], which builds upon the Tensor Field Network formalism [161], Allegro [162], a highly efficient, non-message-passing variant optimized for parallelization [163], and MACE [164, 165, 166].

Message-Passing Atomic Cluster Expansion (MACE)

MACE is a modern class of MLIP that unifies the systematic many-body expansion of the Atomic Cluster Expansion (ACE) with E(3)-equivariant graph neural networks. The central idea is to represent local atomic environments using a hierarchy of many-body geometric features, while ensuring exact equivariance with respect to translations, rotations, and reflections.

In practice, MACE decomposes the total energy into atomic contributions, where each atom interacts with neighbors inside a finite cutoff radius. Information is exchanged between atoms through equivariant message passing, and many-body correlations are incorporated in a controlled and systematic manner via tensor-product feature constructions. This design allows MACE to capture complex local chemistry and geometry with high data efficiency, while preserving the physical symmetries required for accurate energy, force, and stress predictions.

MACE currently achieves state-of-the-art accuracy and transferability across many systems, making it one of the most successful MLIP frameworks to date.

Foundation Models

Finally, a novel approach relies on the development of large-scale foundation models for atomistic simulations, trained on chemically and structurally diverse datasets. Notable examples include SevenNet [168, 169], Orb [170, 171], M3GNet [172], CHGNet [173], GNoME [174], and MACE-MP-0 [175, 176]. These models offer broad transferability across materials classes and enable rapid screening of large chemical spaces.

However, in specialized applications requiring high fidelity, under particular conditions or far-from-equilibrium regimes, their accuracy often remains inferior to that of carefully constructed, system-specific interatomic potentials [177]. While fine-tuning foundation models on targeted datasets can substantially improve performance [177], this process may reduce generality and robustness outside the fine-tuned domain. Consequently, for applications such as phase transitions or high-pressure kinetics, the development of dedicated MLIPs remains favourable.

Chapter 5

Development of a Potential For Germanium

5.1 ML Model Architecture and Training Scheme

This section details the specific architecture, descriptors, loss function, and overall training protocol and hyperparameter selection made in the development of our MLIP, highlighting the rationale for each methodological choice.

Deep Potential Model Hyperparameters

As already mentioned in Sec. 4.2, the model was developed using the Deep Potential Molecular Dynamics (*DeePMD-kit*) package [23, 24]. To adequately represent the diversity in local atomic environments found in the various germanium phases, two types of descriptors were utilized within the so-called *DeepPot-SE* framework: a *se_e3* type embedding that takes angles between two neighboring atoms as input and a *se_e2_a* type embedding that takes the distance between atoms as input; both descriptors are constructed from all the information (both angular and radial). These descriptors are constructed following the procedure outlined in Sec. 4.2 about two and three-body embeddings.

For the angular descriptor, a cutoff $r_c = 3.30\,\text{Å}$ and a smoothing cutoff $r_s = 2.00\,\text{Å}$ were adopted, with the embedding network comprising three layers of 8, 16, and 32 neurons. For the radial descriptor, parameter values of $r_c = 6.60\,\text{Å}$ and $r_s = 3.30\,\text{Å}$ were used, alongside an embedding network with three layers of 16, 32, and 64 neurons and 32 axis neu-

rons. The cutoff radii were selected by checking the radial distribution functions of the structures in the dataset. We ensured that the angular descriptor included at least the first coordination shell, while the radial descriptor encompassed approximately the first three shells (the exact positions of these shells vary depending on the crystal structure and the applied stress conditions). The exact values for the smoothing cutoffs, instead, do not impact models' performance much. The fitting network consisted of three hidden layers containing 160, 120, and 80 neurons, respectively. The sizes of embedding and fitting networks were tuned by starting from educated guesses based on benchmarks shown on the *DeepModeling* documentation [178]. Multiple models were trained and evaluated under varying architectures and hyperparameters to empirically determine the optimal values that balance both high accuracy and speed.

Loss Function and Training Protocol

We trained the DP-model using an Adam stochastic gradient descent optimizer, with an exponentially decaying learning rate. The composite loss function $\mathcal{L}$ incorporates the mean squared errors in energy, forces, and virials:

$$\mathcal{L} = \frac{p_e}{N}\Delta E^2 + \frac{p_f}{3N}\sum_i |\Delta \mathbf{F}_i|^2 + \frac{p_\xi}{9N}\|\Delta \Xi\|^2.$$

Here, ΔE, $\Delta \mathbf{F}_i$, and $\Delta \Xi$ denote the root mean square errors in energy, force, and virial, respectively, calculated between the training data and model predictions; N is the number of atoms, and p_e, p_f, p_ξ are adjustable weights. To accelerate convergence and enhance model transferability, the force weight p_f was made dominant at the initial stages of training and subsequently reduced, while p_e and p_ξ were gradually increased, following recommendations from previous studies [23]. Forces and virials, along with energies, were explicitly included in the training to reduce the quantity of required DFT reference data. Also, such a choice is crucial, in principle, to lead to an interatomic potential capable of better describing pressure-induced phase transitions due to more accurate predictions of stress tensor components.

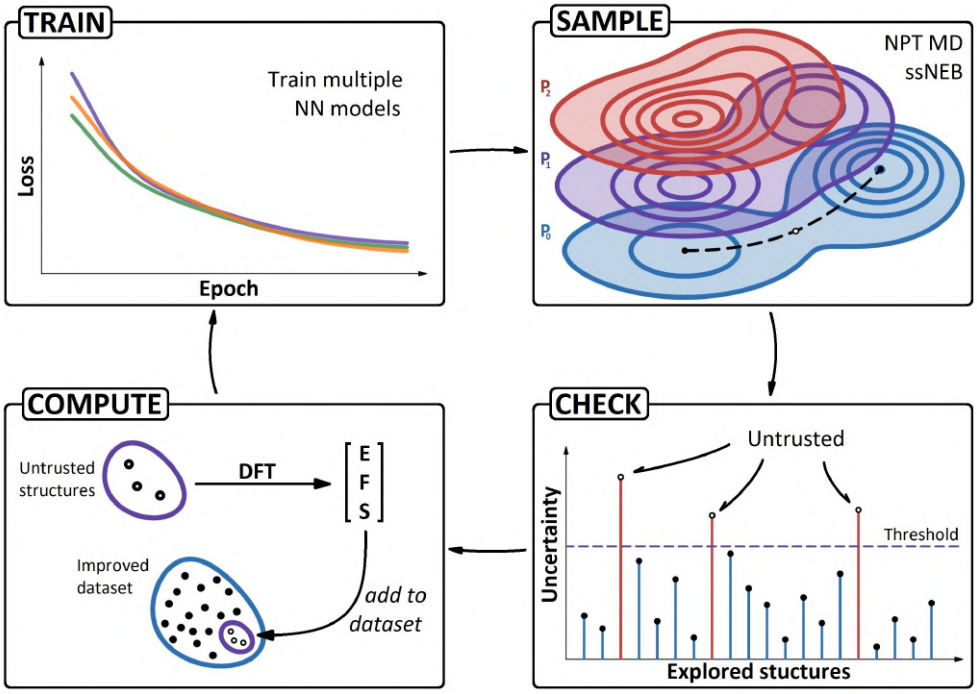

Figure 5.1: Schematic of the active learning workflow we employed. The procedure follows an iterative cycle in which multiple MLIP models are trained. Each model samples a variety of configurations using NPT MD and ssNEB under different stress conditions. The top-right panel illustrates how the enthalpy landscape may shift under applied pressure. The predictions of energies (E), forces (F), and stresses (S) for these sampled structures are compared across the trained models to assess consistency. Configurations showing significant discrepancies are flagged as unreliable and subjected to DFT single-point calculations. These new data points are then added to the training set, and the models are retrained, completing the iterative active learning cycle. Taken from our work, Ref. [179].

5.2 Training Data Curation Through Iterative Active Learning

Active Learning Strategy

The reference dataset for training the MLIP was constructed through an iterative procedure that leverages the principles of active learning [180, 181, 182, 183]. This approach systematically identifies and prioritizes configurations where the current model exhibits unreliable predictions, thereby directing expensive DFT calculations toward regions of configuration space that maximize the information gain per computed structure. This procedure is not new. For example, when constructing surrogate models using Gaussian process regression, the built-in uncertainty estimate (confidence intervals) naturally guides sampling: new points are added in regions of highest predicted uncertainty. This iterative process continues until the domain of interest is sufficiently explored and the estimated uncertainty falls below a chosen threshold.

When a model does not provide intrinsic uncertainty quantification, as is often the case with NN–based MLIPs, alternative strategies are required. A common and effective approach to identify configurations requiring refinement is to train an ensemble of models using identical datasets but initialized with different random seeds. This ensemble approach, often referred to as a model committee, produces diverse predictions for any given configuration. Consistency among the predictions suggests that the structure lies within well-sampled regions of configuration space represented in the training data; conversely, substantial discrepancies among model predictions serve as a useful proxy for uncertainty and help identify where additional sampling is needed. By monitoring the spread in predicted energies, forces, and stresses across the ensemble, configurations exhibiting large deviations were systematically identified, subjected to single-point DFT calculations, and incorporated into the training dataset.

Dataset Construction Workflow

The iterative training protocol, sketched in Fig. 5.1, proceeds according to the following sequence: an initial set of Ge crystalline phases was retrieved from the Materials Project database [184], encompassing CD (*mp-32*), 4H-HD (*mp-1091415*), 2H-HD (*mp-1007760*), BC8 (*mp-1080106*), ST12 (*mp-137*), R8 (*mp-128*), β-Sn (*mp-78*), SH (*mp-1224349*), Imma (*mp-1061054*), Fmmm (*mp-148*), HCP (*mp-1008733*),

Structure type	No. structures	No. environments
Cubic Diam.	253	9 312
Hex. Diam. (4H)	158	6 528
Hex. Diam. (2H)	185	8 016
BC8	173	6 000
ST12	274	8 508
R8	177	6 408
β-Sn	169	5 648
Simple Hex.	150	4 808
Other phases	151	7 280
Interstitials	62	4 092
Vacancies	63	3 906
Disordered	278	26 640
Transitions	597	14 672
TOTAL	$\sim$ 2 700	$\sim$ 112 000

Table 5.1: Summary of the dataset. The columns show the number of structures and the number of atoms (and, therefore, atomic environments) in the database for each structure type.

BCC (*mp-998883*), and FCC (*mp-12093*). Following replication of the primitive unit cells, both lattice parameters and atomic coordinates were subjected to random perturbations to generate structural diversity. These perturbed configurations were then labeled with energies, forces, and stress tensors computed via DFT single-point calculations, thereby establishing the initial training dataset.

A preliminary coarse model trained on this foundational dataset was subsequently employed to explore additional regions of configuration space through NPT and NVT molecular dynamics simulations, as well as through the extraction of intermediate randomly perturbed images from ssNEB calculations connecting different crystalline phases. The energies, forces, and stresses of these newly sampled configurations were evaluated via DFT and incorporated into the expanding dataset.

Dataset Details

The final curated dataset comprises approximately 2700 structures, encompassing a total of roughly 112, 000 distinct atomic environments, which were randomly partitioned into 87% training and 13% validation

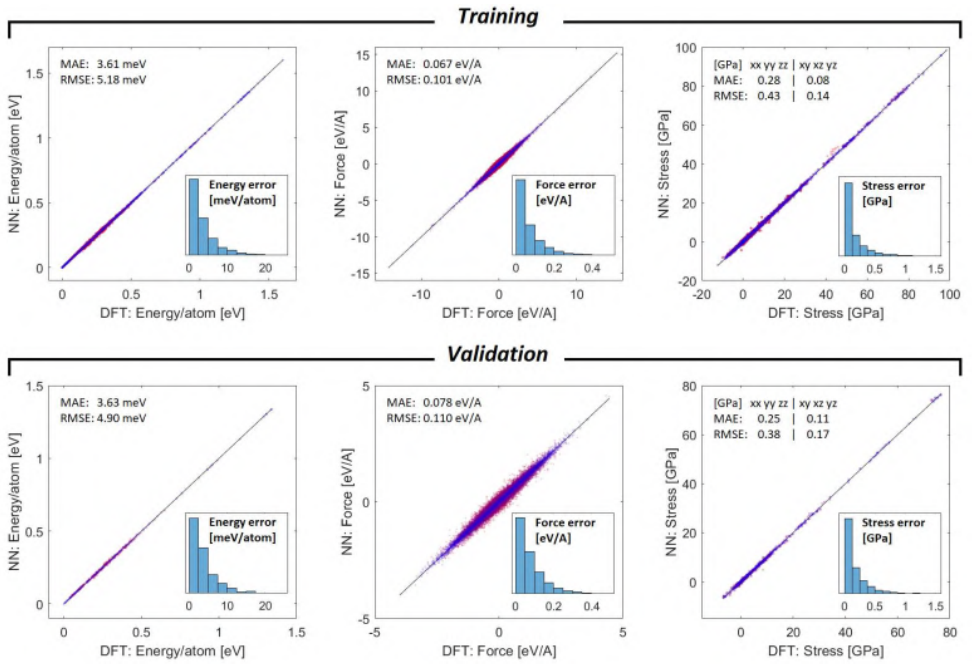

Figure 5.2: Regression plots made on training and validation sets, illustrating correlations of energies, forces, and stresses predicted by the trained deep neural network (NN) potential with corresponding DFT values. Adapted from our work, Ref. [179].

subsets.

Given that the phase transformations of interest are predominantly pressure-induced, the dataset encompasses structures spanning an extensive pressure range, reaching nearly 100 GPa for certain configurations. The majority of sampling efforts were concentrated within a pressure window extending from 5 GPa tensile stress to 30 GPa compressive stress. This range was selected based on the observation that technologically relevant transformations in germanium, including the formation of hexagonal diamond and ST12 phases following compression of CD to β-Sn and subsequent decompression, typically occur below 20 GPa in experimental investigations.

Defective configurations containing interstitial atoms and vacancies were generated by sampling molecular dynamics trajectories initiated from manually constructed defect structures. Disordered and amorphous-like configurations were obtained through heating-cooling cycles applied to selected crystalline phases, producing structures with substantial local disorder. These structurally varied configurations facilitate compre-

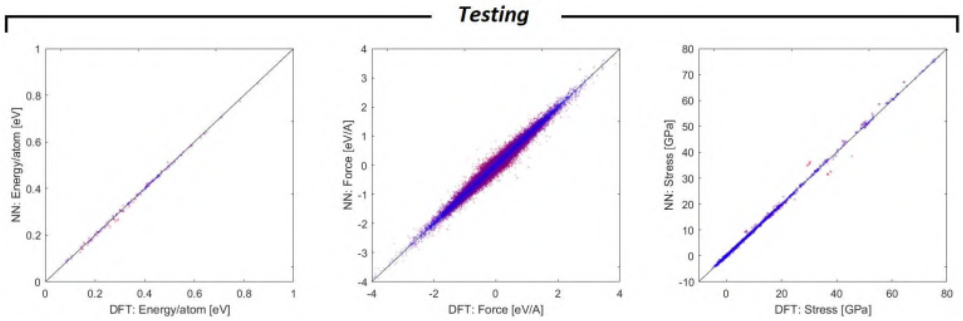

Figure 5.3: Regression plots made on the test set, illustrating correlations of energies, forces, and stresses predicted by the trained deep neural network (NN) potential with corresponding DFT values. Adapted from our work, Ref. [179].

hensive sampling of diverse atomic environments, thereby enhancing the robustness and transferability of the trained potential.

The comprehensive composition of the resulting dataset is summarized in Tab. 5.1, delineating the number of structures and atomic environments associated with each structural category: bulk crystalline phases (CD, 4H-HD, 2H-HD, BC8, ST12, R8, β-Sn, SH, and other phases), point defects (interstitials and vacancies), disordered configurations, and structures sampled from transition pathways.

5.3 Benchmarking and Validation of the Model

Training Outcomes and Model Assessment

Performance of the trained model was evaluated through analysis of regression plots on both training and validation sets, shown in Fig. 5.2. The final model demonstrated root mean square errors (RMSEs) of approximately 5 meV for energies and 0.1 eV/Å for interatomic forces. Stress predictions were robust, with RMSE values consistently below 0.5 GPa across a spectrum extending up to 100 GPa.

To ensure the reliability of the constructed MLIP for germanium, exhaustive benchmarking was conducted against first-principles reference data across a broad spectrum of structures and atomic environments. This comprehensive evaluation establishes the robustness of the DP-model and its suitability for probing phase transformation mechanisms.

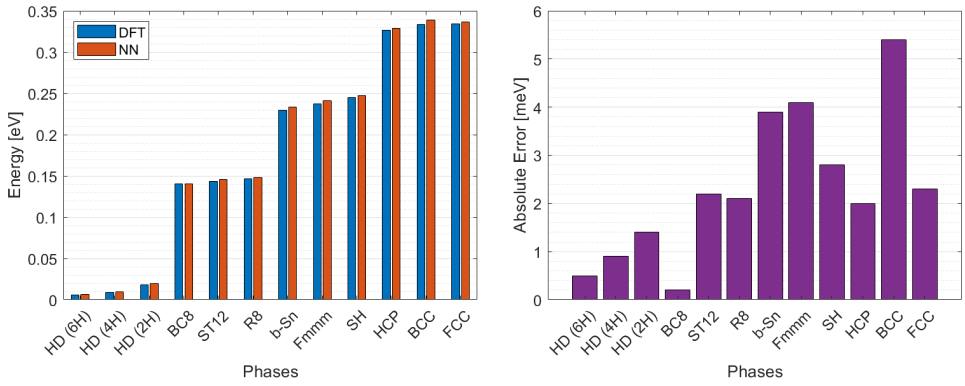

Figure 5.4: Comparison of formation energies for various crystalline Ge phases, computed using DFT (blue) and the developed MLIP (red). Adapted from our work, Ref. [179].

Test Set Composition and Predictive Performance

A diverse test set was meticulously assembled, comprising snapshots from MD simulations performed at differing temperatures and pressures. The test set included 125 diamond-phase structures (CD, 4H-HD, 2H-HD, totaling 7888 atomic environments), 74 low-density semiconductor structures (BC8, ST12, R8; 3984 atomic environments), 97 high-density metallic phases (6144 environments), and 146 disordered or defective structures (spanning amorphous, interstitial, and vacancy configurations, totaling 13,401 atomic environments). Even on this test set, the DP-model displayed satisfactory predictive accuracy, with reasonable deviations from DFT values in regression analyses of energies, forces, and stresses, shown in Fig. 5.3.

Formation Energy Assessment

Formation energies were computed as the per-atom energy difference between an optimally relaxed configuration and the cubic diamond reference, both at the DFT and model-predicted level. Across all tested crystal phases, including those typical of high-pressure and metastable behavior, the DP-model reproduced reference formation energies with high fidelity—absolute errors generally below 5 meV, except for the BCC phase, which exhibited slightly larger deviations, as shown in Fig. 5.4.

The DP-model performed better on prominent low-energy phases such as 4H-HD, 2H-HD, BC8, ST12, and R8, demonstrating errors smaller than those observed in the high-energy metallic phases. Notably, the

Phase	$\varepsilon_{\mathrm{DFT}}$ [meV]	$\varepsilon_{\mathrm{DP}}$ [meV]	$\Delta\varepsilon$ [meV]
6H-HD	5.7	6.2	0.5
4H-HD	8.6	9.5	0.9
2H-HD	18.0	19.4	1.4
BC8	140.5	140.7	0.2
ST12	143.7	145.9	2.2
R8	146.3	148.4	2.1
β-Sn	229.3	233.2	3.9
Fmmm	237.4	241.5	4.1
SH	244.8	247.6	2.8
HCP	326.8	328.8	2.0
BCC	333.6	339.0	5.4
FCC	334.6	336.9	2.3
Pbam	31.8	56.7	24.9
$P4_2/ncm$	35.8	32.7	-2.9
$P4_12_12$	38.7	73.9	35.2

Table 5.2: Formation energies (ε) and absolute error ($\Delta\varepsilon$) computed by DFT and the DP-model for different crystalline phases of Ge.

accurate prediction of the 6H-HD phase was achieved despite its absence in the training dataset, attributed to the inclusion of structurally related phases (CD, 4H-HD, and 2H-HD), providing sufficient coverage of local atomic environments.

We tested the DP-model also on three additional phases, which were not part of the training set: Pbam, $P4_2/ncm$, $P4_12_12$. These recently predicted structures exhibit lower energies than most known metastable polymorphs of Ge. For this reason, it has been suggested that they could potentially appear as metastable phases in experiments employing indentation or anvil-cell techniques [185, 186]. As shown in Tab, 5.2, the comparison between DFT formation energies and DP predictions reveals noticeably larger errors for these configurations, which the model hasn't been trained on. Any study focusing on these structures would certainly require incorporating them into a newly refined training set to achieve reliable accuracy.

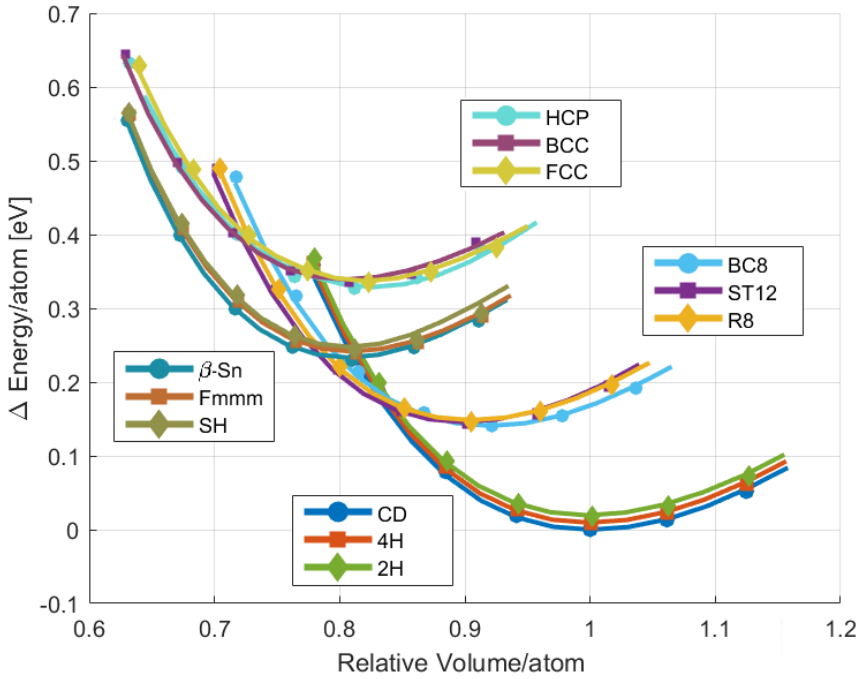

Figure 5.5: Comparison of formation energies for various crystalline phases of Ge calculated using DFT (blue) and the developed MLIP (red). Taken from our work, Ref. [179].

In addition, we analyzed the accuracy of energy-volume curves across crystalline phases to assess elastic response, as shown in Fig. 5.5. The regression results showed exceptional correspondence between model and DFT for diamond-like phases and low-density semiconducting phases. For metallic phases, some decline in accuracy was found for volumetric expansion regimes, but the overall trend is still well-reproduced. Such behaviour is expected due to a lack of representation of those expanded structures, as the configurations of interest are in pressure-induced compression scenarios.

5.4 Minimum Energy Transition Pathways

With the accuracy of the MLIP established on reference data, we can evaluate its capability to predict barriers and pressure-induced transition pathways between different Ge phases. Here, we systematically detail the ssNEB calculations we employed under various stress conditions.

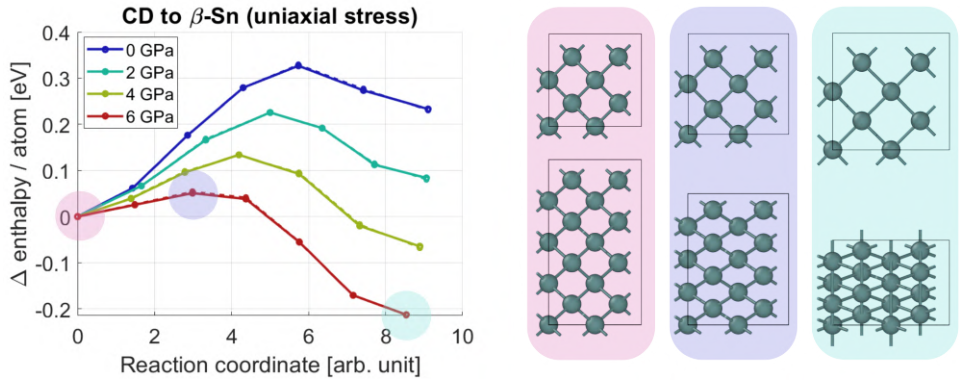

Figure 5.6: (left) Minimum energy paths of the phase transition CD to β-Sn under uniaxial stress. A comparison between the ssNEBs calculations performed with the DP model (solid lines and filled circles) and single-point DFT calculations of such images (dashed lines and unfilled circles) is shown. (top-right) Top views and (bottom-right) side views of a representative 16-atom cell depicting the transition from left to right from CD, through the saddle-point configuration, and to β-Sn, at 6 GPa of uniaxial compression along the (100)-direction, here ortogonal to the plane of the top view.

CD to β-Sn

The analysis began with an examination of the transformation between cubic diamond (CD) and β-Sn. This transformation exhibits a particularly straightforward behaviour: the relative positions of atoms within the unit cell remain unchanged, while only the lattice parameters are modified. This crystal phase transformation, already known for silicon [187], is present in the same way in germanium. The computed activation energy for this transition stands at approximately 0.32 eV/atom, representing a significant kinetic barrier under ambient conditions. However, the application of external stress can significantly alter such a barrier.

Under stress, the relevant thermodynamic potential shifts from energy to enthalpy, taking into account the work performed by applied forces. To quantify this effect, a series of ssNEB calculations was performed across a range of values of uniaxial compressive stress. These calculations revealed that at approximately 7 GPa of uniaxial compression along the (100)-direction, the activation barrier vanishes entirely, rendering the CD-to-β-Sn transformation thermodynamically spontaneous. The ssNEB energy profile computed using the ML potential exhibited

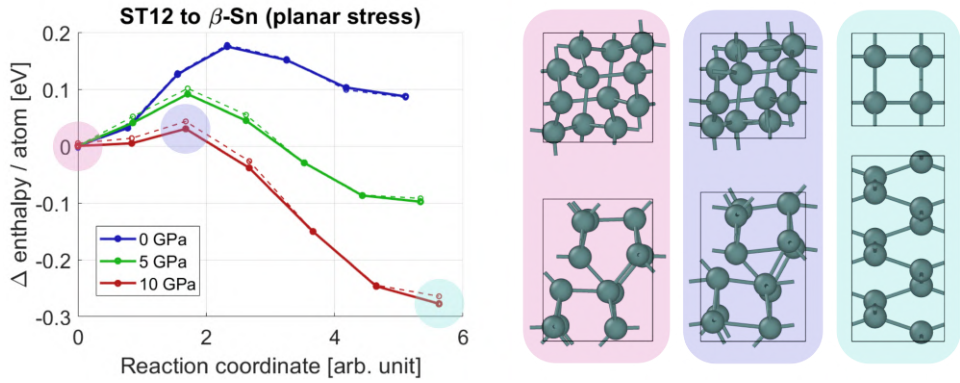

Figure 5.7: (left) Minimum energy paths of the phase transition ST12 to β-Sn under planar stress. A comparison between the ssNEBs calculations performed with the DP model (solid lines and filled circles) and single-point DFT calculations of such images (dashed lines and unfilled circles) is shown. (top-right) Top views and (bottom-right) side views of a representative 12-atom cell depicting the transition from left to right from ST12, through the saddle-point configuration, and to β-Sn, at 10 GPa of planar compressive stress along the plane here parallel to the top view plane.

remarkable quantitative agreement with single-point DFT calculations of the intermediate images, as shown in Fig. 5.6.

ST12 to β-Sn

Further investigations targeted the transformation connecting ST12 and β-Sn phases. Employing a 12-atom simulation cell, the computed minimum energy pathway aligned closely with previously proposed atomistic mechanisms for this transition [188]. The activation barrier in this case was estimated at 0.18 eV/atom. Once again, we observe that the transition requires overcoming a relatively high barrier under ambient conditions. However, we also find that applying appropriate stress significantly lowers this barrier and alters the relative enthalpic stability of the two phases. Specifically, the imposition of planar compressive stress systematically reduced the barrier height, ultimately inducing a barrierless transition at approximately 12 GPa of applied stress, as illustrated in Fig. 5.7.

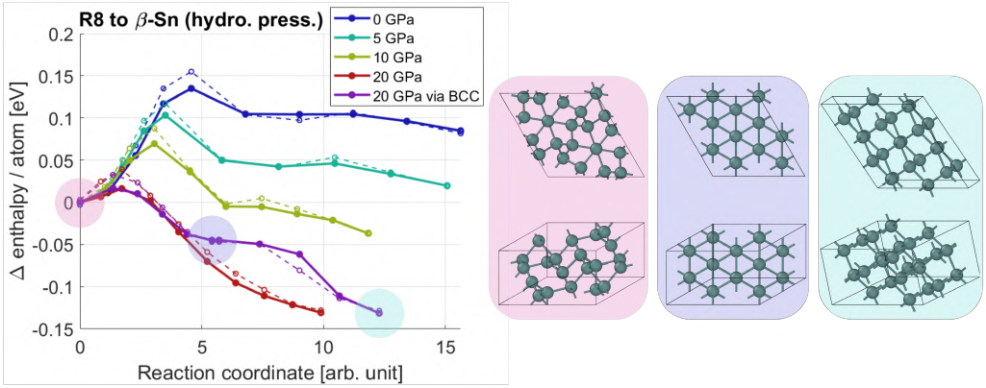

Figure 5.8: (left) Minimum energy paths of the phase transition R8 to β-Sn under hydrostatic pressure. A comparison between the ssNEBs calculations performed with the DP model (solid lines and filled circles) and single-point DFT calculations of such images (dashed lines and unfilled circles) is shown. (top-right) Top views and (bottom-right) side views of a representative 12-atom cell depicting the transition from left to right from R8, via the BCC phase, and to β-Sn, at 20 GPa of hydrostatic stress.

R8 to β-Sn

As a last case study, the influence of hydrostatic pressure on the R8-to-β-Sn transformation was also examined. The computed energy profiles indicated a progressive reduction in the activation barrier with increasing pressure, as illustrated in Fig. 5.8. Notably, above 20 GPa, a qualitatively distinct multistep pathway emerged, proceeding through a transient BCC intermediate phase. To accurately resolve this complex pathway, additional intermediate images were incorporated into the ssNEB chain, with separate calculations performed between successive local minima along the reaction coordinate.

A comparison between the MLIP-predicted energy profiles and DFT single-point evaluations for the intermediate structures along the R8-to-β-Sn pathway revealed notable discrepancies, with prediction errors approaching 30 meV at certain reaction coordinates. This low accuracy arises in this case because the R8-to-β-Sn transition was investigated only after the completion of model training, and consequently, intermediate configurations along this specific pathway were absent from the training dataset. Once again, this observation underscores a fundamental limitation of machine learning potentials: their predictive reliability is restricted to regions of configuration space adequately sampled dur-

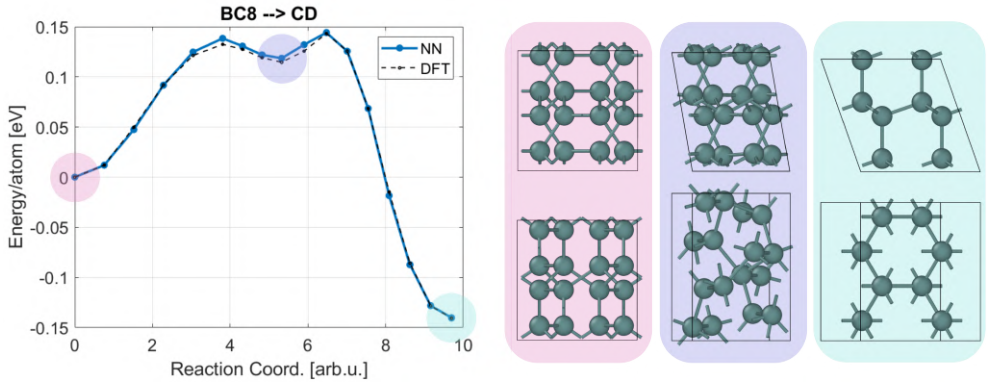

Figure 5.9: (left) Minimum energy path of the phase transition BC8 to CD. A comparison between the ssNEBs calculations performed with the DP model (solid lines and filled circles) and single-point DFT calculations of such images (dashed lines and unfilled circles) is shown. (top-right) Top views and (bottom-right) side views of a representative 16-atom cell depicting the transition from left to right from BC8, through an intermediate shallow minimum, to the CD phase.

ing training, highlighting the necessity of incorporating transition-state structures for pathways of interest.

BC8 to cubic and hexagonal diamond competing pathways

Depending on the choice of the simulation cell, its geometry and orientation, and the atom-to-atom correspondence between the initial and final cells used in the ssNEB calculations, different activation barriers can be obtained. Naturally, our primary interest lies in identifying the lowest barrier, as it corresponds to the most probable transition mechanism and is therefore most relevant to the actual physical process. We initially identified a very high energy barrier, employing a 16-atom simulation cell, and featuring a shallow metastable intermediate configuration, as shown in Fig. 5.9.

As an alternative approach, we performed multiple dimer searches starting from the BC8 phase, in order to explore a broader set of possible escape pathways. Once the dimer converges to a saddle point, the two associated images are minimized to locate the corresponding minima connected by that saddle. These minima, together with the saddle point and interpolated intermediate images, are then used as an initial guess for subsequent ssNEB calculations. By repeating this procedure several times, we identified two competing transition pathways, BC8 $\rightarrow$ CD and

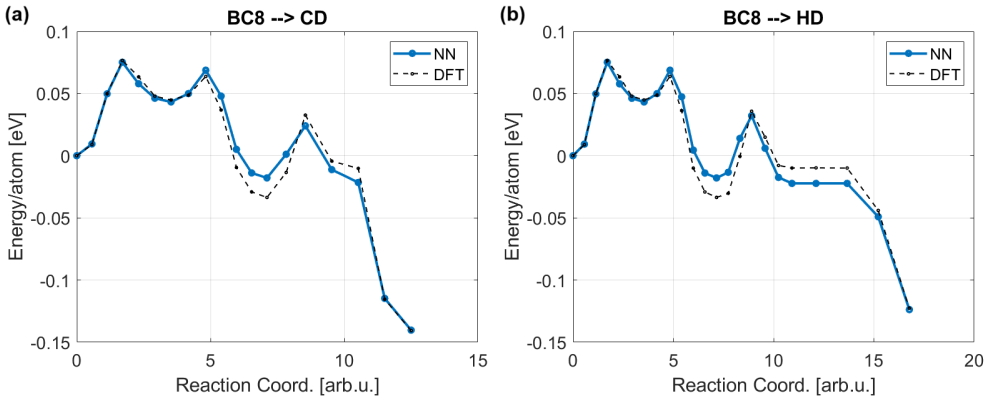

Figure 5.10: Comparison between the minimum energy paths, employing a 16-atom cell, of the phase transitions (a) BC8 to CD and (b) BC8 to HD. Also, a comparison between the ssNEBs calculations performed with the DP model (solid lines and filled circles) and single-point DFT calculations of such images (dashed lines and unfilled circles) is shown.

BC8 → HD. Interestingly, they share the same initial segment of the reaction path, which also determines the highest crossing point in energy, as illustrated by comparing Figs. 5.10a and 5.10b.

In contrast to the ssNEB calculations discussed in the previous sections, where, by applying different stress states, we allowed the transition barriers to be significantly reduced or even eliminated, the NEB results for the BC8 → CD and BC8 → HD transitions are less informative. The computed barriers remain high, while the BC8 → HD transition is instead observed to occur spontaneously during annealing [76, 77], without the need to modify the transition rate through applied pressure. Moreover, the reason why formation of HD is preferred rather than the slightly more energetically favourable CD phase is still not understood, given these results. As a consequence, these NEB calculations provide limited insight into the actual transition mechanism, which is more likely governed by nucleation processes. A more thorough investigation, similar to that performed for silicon in Ref. [189], would therefore be required to accurately capture the underlying physics of the transition. Such an investigation could clarify why a spontaneous BC8 → HD transition is not observed, possibly invoking the same rationale proposed in Ref. [189]: the non-uniform distribution of residual stresses in a typical nanoindentation experiment lowers the barrier to start the nucleation of the HD phase inside a BC8 matrix.

5.5 Pressure-Dependent Kinetics

To complement the ssNEB analysis and provide direct dynamical insight into pressure-induced phase transformations, NPT molecular dynamics simulations were performed. These simulations enable observation of nucleation and growth processes at finite temperature, providing a more realistic picture of transformation kinetics.

CD to β-Sn MD simulation

The first case examined was the CD-to-β-Sn transformation under uniaxial compressive stress. A simulation cell with approximate dimensions $56 \times 56 \times 76$ Å, containing 10,368 atoms, was employed. The system was maintained at room temperature (300 K) with uniaxial compressive stress applied along the (100)-direction at 7 GPa, the stress magnitude identified from ssNEB calculations in the previous section as sufficient to eliminate the activation barrier. Snapshots extracted from the resulting MD trajectory are displayed in Fig. 5.11a of the original work, illustrating the progression of the structural transformation. Color coding based on coordination analysis clearly distinguishes the CD (red) and β-Sn (blue) phases, revealing the transition from one phase to the other.

To assess the reliability of the MLIP during this dynamical transformation, the deviation in predicted atomic forces was monitored throughout the simulation using an ensemble of three independently trained DP-models with different random initialization seeds. This is a viable option commonly used when one wants to quantify at least a rough estimate of the uncertainty of the model predictions. At each timestep, the mean force deviation across all atoms was computed, and the maximum deviation among individual atoms was recorded. As shown in Fig. 5.11b, force predictions remained mostly quite accurate throughout the transformation process, with mean deviations consistently below 0.05 eV/Å and peak deviations not exceeding 0.15 eV/Å, even at the transition midpoint where atomic environments deviate more from equilibrium configurations. These low error magnitudes confirm that the DP-model maintains satisfactory predictive accuracy across the full range of structural configurations encountered during the phase transition.

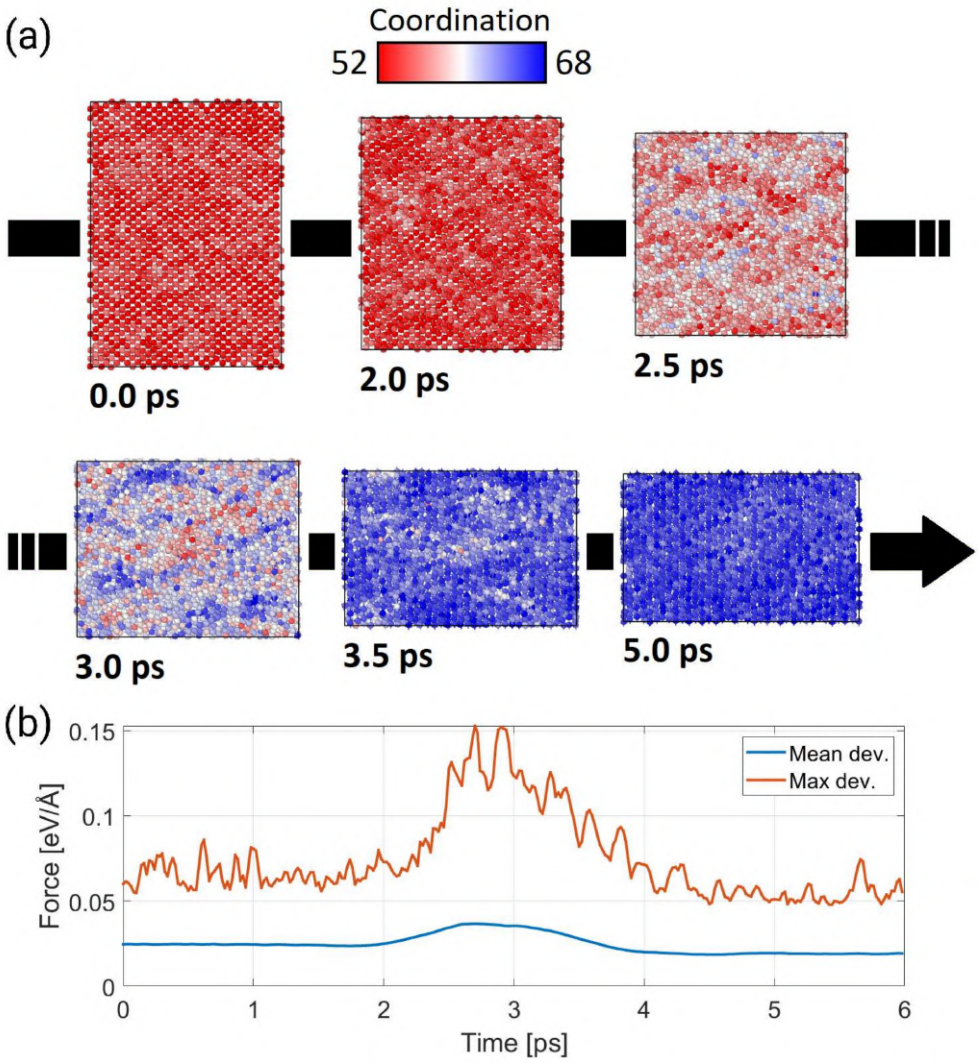

Figure 5.11: (a) Representative snapshots of slices extracted from an NPT MD trajectory of the pressure-induced transformation from the CD phase to β-Sn. The simulation is performed at 300 K under uniaxial compressive stress applied along the (100)-direction at 7 GPa. Atoms are colored according to coordination analysis performed with Ovito [190], with the CD and β-Sn phases shown in red and blue, respectively. (b) Evolution of the average and maximum force deviations during the MD simulation. Adapted from our work, Ref. [179].

R8 to β-Sn MD simulation

A second NPT-MD simulation targeted the R8-to-β-Sn transformation under hydrostatic compression. A rhomboidal prism-shaped simulation cell containing 12,288 atoms was constructed, with the system maintained at 200 K and subjected to hydrostatic pressure of 23 GPa. This pressure value was selected based on the ssNEB calculations in the previous section, which indicated a kinetic barrier of approximately 8 meV/atom at this condition: sufficiently low to permit direct observation of the transformation on accessible MD timescales.

Snapshots captured during the simulation are shown in Fig. 5.12a, interestingly revealing the nucleation of the BCC intermediate phase within the R8 bulk, followed by its subsequent transformation into polycrystalline β-Sn with residual defects. The color-coded visualization based on coordination analysis clearly distinguishes the initial R8 structure (blue), the transient BCC nuclei (red), and the final β-Sn product (blue again). This observation provides direct computational evidence for the multistep transformation pathway identified in ssNEB calculations at pressures exceeding 20 GPa in Sec. 5.4.

Force prediction accuracy during this simulation, shown in Fig. 5.12b, exhibits mean deviations of approximately 0.05 eV/Å during the initial nucleation stages, decreasing below 0.02 eV/Å as the transformation progresses and the system evolves toward the β-Sn product. The improved accuracy in later stages reflects the fact that β-Sn configurations were extensively sampled in the training dataset, whereas transient BCC structures, though present, were less comprehensively represented. More specifically, Fig. 5.12b shows a clear decrease in the mean force deviation between 12 and 14 ps, after which it stabilizes as the β-Sn phase forms. In contrast, the maximum force deviation continues to fluctuate around an approximately constant value of ≈ 0.1 eV/Å from about 8 ps onward, reflecting the persistent presence of defects in the system.

It is noteworthy that while the reverse transformation (from β-Sn to R8 upon decompression) has been experimentally observed in nanoindentation experiments under specific unloading rates and stress conditions, the forward R8-to-β-Sn transition under loading has received considerably less experimental attention, primarily because producing R8 from β-Sn is a necessary prerequisite. Consequently, direct experimental benchmarks for this transformation are scarce. Nonetheless, the computational predictions presented here may prove valuable for interpreting cyclic nanoindentation experiments and understanding the hysteresis in phase transformation pathways, like those in Ref. [70].

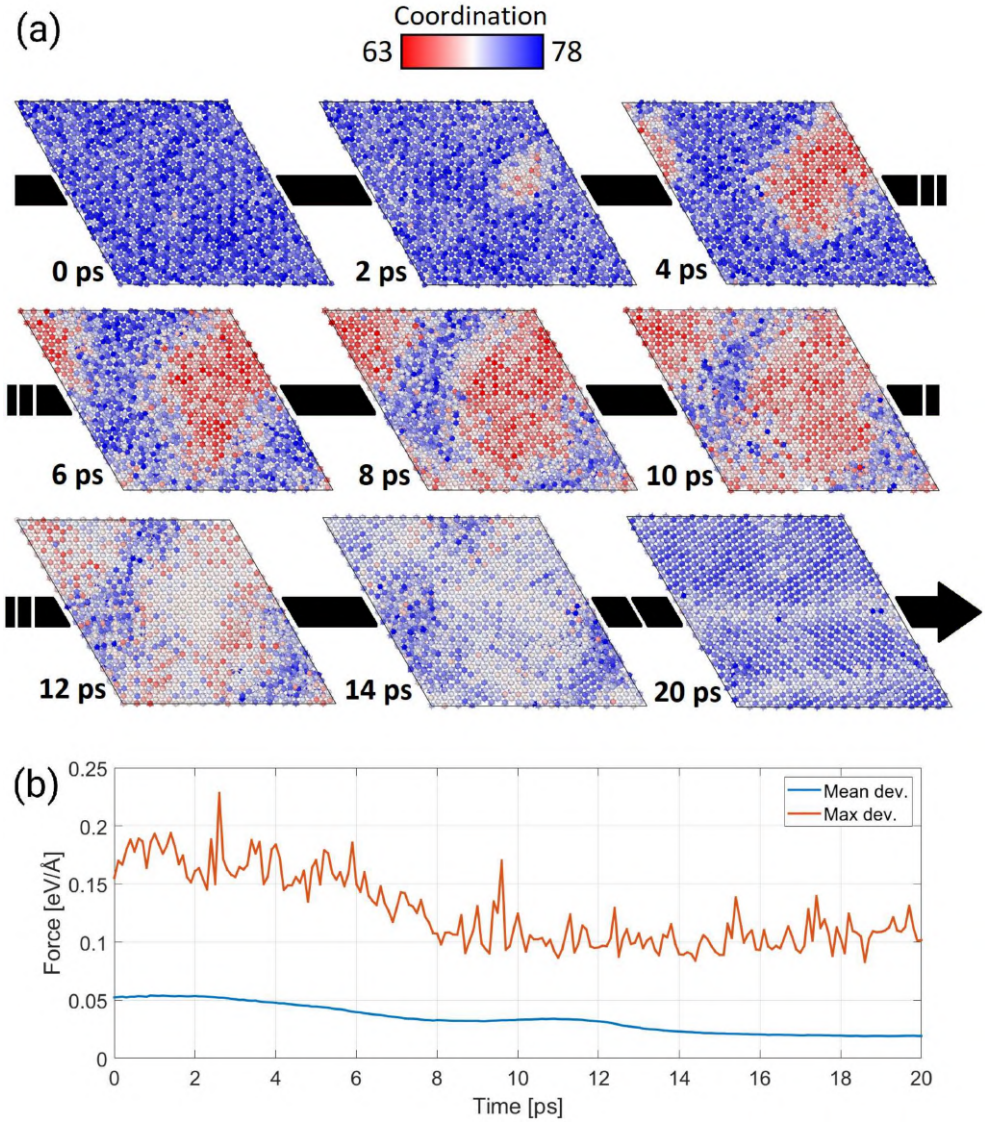

Figure 5.12: (a) Representative snapshots of slices extracted from an NPT MD trajectory illustrating the pressure-induced transformation from the R8 phase to β-Sn through a BCC intermediate. The simulation is carried out at 200 K under a hydrostatic pressure of 23 GPa. Atoms are colored according to coordination analysis performed in Ovito, highlighting first the R8 matrix (blue), the nucleating BCC region (red), and the final transformation to a polycrystalline β-Sn phase (blue). (b) Time evolution of the average and maximum force deviations during the MD simulation. Adapted from our work, Ref. [179].

Summary of achievements, model limits, and implications

The close agreement between ssNEB energy profiles computed with the developed MLIP and reference DFT calculations for many crystal phases and transitions, such as CD to β-Sn and ST12 to β-Sn, demonstrates the model's ability to achieve near–DFT accuracy while offering computational efficiency several orders of magnitude higher than first-principles methods. This capability provides a powerful computational framework for interpreting and complementing experimental investigations of pressure-induced transformations.

Despite these successes, some limitations remain and point to directions for future work. Elevated prediction errors observed for theoretically proposed crystal phases not included in the training set (e.g., Pbam, $P4_2/ncm$, $P4_12_12$) indicate that extending the model's applicability requires deliberate inclusion of such structures during training. Similarly, the R8 to β-Sn pathway, also examined in detail only after the model had been trained and therefore lacking representative intermediate configurations in the training set, exhibits discrepancies of up to 30 meV at intermediate images. This result emphasizes the critical role of explicitly sampling transition-relevant configurations when constructing robust and transferable interatomic potentials.

To facilitate reproducibility and further research, the developed MLIP, together with the full reference dataset, has been made publicly available at Ref. [191].

PART III
Continuum Modeling

Chapter 6

Theory & Methods for Continuum Modeling

6.1 Context & Motivation

Studying the time evolution of microstructural features in materials is fundamental to materials science and engineering, as these features often determine functional and mechanical properties [192, 193]. To simulate these complex dynamics, continuum models, particularly phase-field (PF) methods, have been developed and refined over decades [14]. These approaches can, however, be demanding when one wants to perform high-throughput studies, large-scale domain simulations, or long-time evolution predictions.

The main goal we aim for is to build a surrogate model, based on a modern NN architecture, capable of accelerating continuum-scale materials simulations while maintaining accuracy comparable to conventional direct numerical solvers (DNS). In recent years, indeed, NNs have demonstrated their ability to achieve substantial speed-ups, frequently reported in the order of $100 - 1000\times$ compared to these solvers [17, 15, 16].

To build such surrogate models, approaches based on dimensionality reduction [18, 194, 195], convolutional NNs (CNNs) [196, 197, 16, 198] and various modern time-series specific architectures such as recurrent NNs (RNNs) [19, 20] have been employed. A comprehensive overview of these powerful approaches is given in the next chapter.

In the following section, we begin by describing the PF model of coherently strained spinodal decomposition: a prototypical system we selected because it displays physically meaningful, complex microstructural evolution while still allowing for a relatively simple numerical implemen-

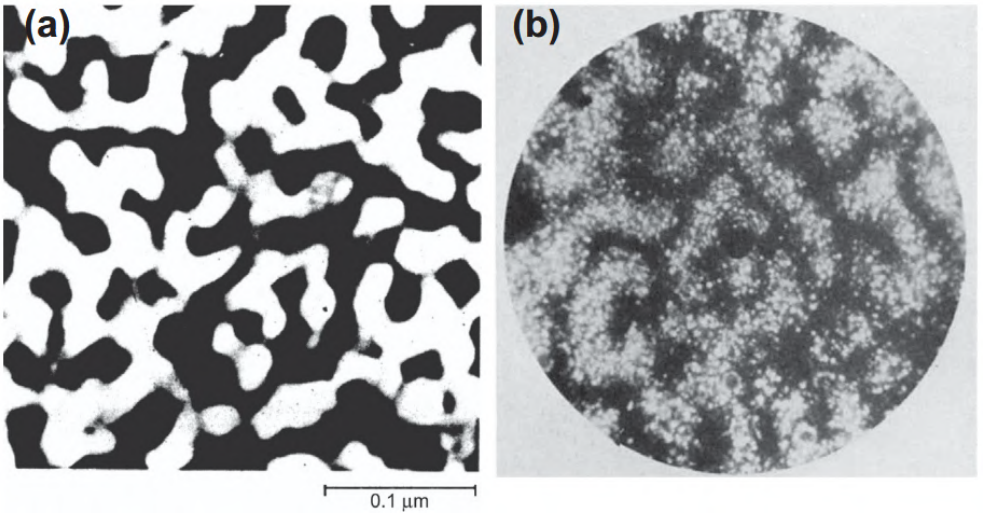

Figure 6.1: (a) TEM image of an isotropic spinodal morphology developed in an Fe–Cr–Co alloy (Zeltzer). (b) Field ion micrograph image of an isotropic spinodal morphology in an Fe–Cr–Co alloy. Taken from Ref. [199] and Ref. [200] with permission.

tation. After that, the dataset generation process that provides training data for our NN approach is discussed.

6.2 Phase Field Model of Coherently Strained Spinodal Decomposition

The phenomenon of spinodal decomposition represents a prototypical second-order phase transition wherein a homogeneous mixture spontaneously segregates into two distinct phases upon cooling below a critical temperature. This process is particularly relevant in certain crystalline alloy systems where the two different emerging phases exhibit different molar volumes, leading to the development of elastic strain fields that influence the resulting microstructural morphologies. An example of this phenomenon is given in Fig. 6.1.

Thermodynamic Framework for Spinodal Decomposition

The modeling framework employed here adopts a PF approach [14] that represents the system through an order parameter φ tracking the local composition of the binary alloy, coupled with the total strain ten-

sor field $\varepsilon(\mathbf{x})$. Under isothermal conditions and assuming linear elasticity, the system's free energy is expressed through the Ginzburg-Landau functional [201]

$$F[\varphi, \varepsilon] = \int_{\mathcal{D}} \left(\frac{\epsilon}{2} |\nabla \varphi|^2 + w(\varphi) + \rho(\varphi, \varepsilon) \right) d\mathbf{x} \tag{6.1}$$

where $\mathcal{D}$ denotes the physical domain and ϵ characterizes the interface width between phases. The gradient term $|\nabla \varphi|^2$ penalizes interfacial energy, while the bulk contribution consists of a double-well potential

$$w(\varphi) = \frac{18}{\epsilon} \varphi^2 (1 - \varphi)^2 \tag{6.2}$$

featuring energy minima at $\varphi = 0$ and $\varphi = 1$, corresponding to the equilibrium compositions of the separated phases without strain. Note that the effect of temperature is taken into account in an implicit way. We decided to assume a fixed temperature so that the physical composition c in the range $[c_1,\ c_2]$ is rescaled to the PF composition $\varphi = (c - c_1)/(c_2 - c_1)$.

Microstructure Evolution and Elastic Effects

The elastic energy density $\rho(\varphi, \varepsilon)$ accounts for lattice mismatch between the constituent materials:

$$\rho(\varphi, \varepsilon) = \frac{1}{2} C_{ijkl} (\varepsilon_{kl} - \varepsilon^*_{kl}(\varphi))(\varepsilon_{ij} - \varepsilon^*_{ij}(\varphi)) \tag{6.3}$$

where C_{ijkl} represents the fourth-order elastic constant tensor and ε^* is the eigenstrain tensor (zero-stress strain) [202]. For elastically homogeneous materials with purely dilatational eigenstrain:

$$\varepsilon^*_{ij}(\varphi) = \eta(\varphi - \langle \varphi \rangle)\delta_{ij} \tag{6.4}$$

where η denotes the lattice mismatch parameter and $\langle \varphi \rangle$ represents the mean composition.

Since mechanical relaxation occurs much faster than diffusive phase separation, a quasi-static equilibrium assumption allows the strain field to be computed directly [203]. For periodic, elastically homogeneous systems, this is efficiently achieved in Fourier space, where the equilibrium stress field $\hat{\boldsymbol{\sigma}}$ relates to the composition field as:

$$\hat{\boldsymbol{\sigma}} = -\eta \mathbf{B}(\mathbf{q}) \hat{\varphi} \tag{6.5}$$

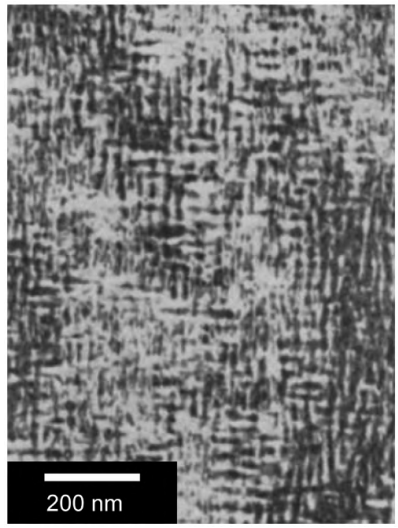

Figure 6.2: Spinodal microstructure revealed by TEM in an aged CuNiFe alloy showing a periodic, crystallographically aligned two-phase mixture; the foil normal is approximately [001], and the particles of the second phase are aligned along the [100] and [010] matrix directions. Adapted from Ref. [206] with permission.

$$B_{ij}(\mathbf{q}) = C_{ijkl}(\delta_{kl} - \Omega_{km} C_{mnop} \delta_{op} q_n q_l) \tag{6.6}$$

with $\mathbf{q}$ representing the wavevector and $\mathbf{B}$ a tensor depending on elastic constants and the acoustic Green's function.

The temporal evolution follows the Cahn-Hilliard equation:

$$\frac{\partial \varphi}{\partial t} = M \nabla^2 \mu = M \nabla^2 \left(-\epsilon \nabla^2 \varphi + w'(\varphi) + \rho'(\varphi, \varepsilon) \right) \tag{6.7}$$

where $\mu = \delta F / \delta \varphi$ is the chemical potential, M denotes mobility, and primes indicate derivatives with respect to φ. Numerical integration employs a semi-implicit spectral scheme for computational efficiency.

Linear stability analysis reveals that elastic strain generally stabilizes the homogeneous phase, extending the metastability region within the spinodal one. For materials with cubic symmetry and anisotropic elastic properties, which we are focusing on, phase separation becomes direction-dependent, potentially leading to suppression along crystallographically hard directions and enhancement along soft orientations, resulting in characteristic anisotropic morphologies such as striped domains [204, 205]. An example of this phenomenon is given in Fig. 6.2.

Numerical Implementation of the Phase-Field Model

The numerical solution of the phase-field model for coherently strained spinodal decomposition is based on efficiently integrating the time-dependent Cahn-Hilliard equation coupled with elasticity in two-dimensional domains. The time evolution of the compositional order parameter φ is computed using a semi-implicit spectral scheme [207]. This approach balances computational cost and stability, allowing for accurate simulation of microstructure dynamics.

The numerical equation for the evolution, written in Fourier space, is [202, 203, 208]:

$$\widehat{\varphi}_{t+\delta t} = \frac{\widehat{\varphi}_t - \delta t M q^2 \widehat{w'(\varphi_t)}}{1 + \delta t M \left(\epsilon q^4 + \eta^2 \mathrm{tr}(\mathbf{B}) q^2\right)} \tag{6.8}$$

where $\widehat{\varphi}$ and $\widehat{w'}$ are the Fourier transforms of the composition field and the derivative of the double-well potential, δt is the time step, M is the mobility, ϵ the interface energy coefficient, η the misfit strain, and B is the elastic modulus-dependent matrix from the equilibrium stress solution. This implementation works particularly well thanks to the use of fast Fourier transforms for efficient computation.

Lastly, the onset of phase separation and the stability of the homogeneous state can be determined through a linear stability criterion [208]:

$$w''(\varphi) + \eta^2 \min_{\mathbf{q}} \left[\mathrm{tr}(B(\mathbf{q}))\right] < 0 \tag{6.9}$$

This condition shows that elastic strain can suppress decomposition, stabilizing the homogeneous phase and introducing anisotropic effects dependent on crystallographic orientation and strain magnitude.

In this work, we focus on an arbitrary material with a high (cubic) Zener anisotropy ratio $Z = 2C_{44}/(C_{11} - C_{12}) = 4$ and a misfit range $\eta \in [0, 1.2\%]$. Elastic constants are $C_{11} = C_{44} = 2C_{12} = 3 \cdot 10^4$. With this choice of values, we aim at maximizing the variability in the phase-separation patterns. Following this choice of parameters, softer directions are aligned with the $\langle 10 \rangle$ and $\langle 01 \rangle$ axes so that, for high strains, striped domains are expected to form along them. For this choice of parameters, the resulting phase diagram, together with representative examples of spinodal decomposition morphologies, is shown in Fig. 6.3.

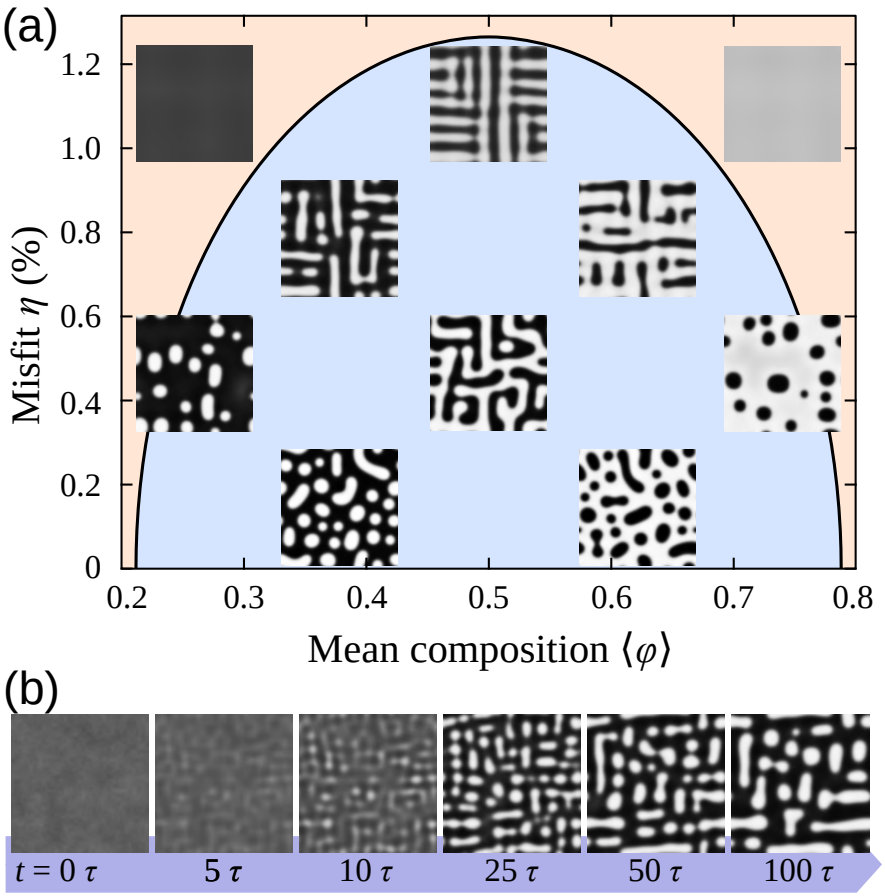

Figure 6.3: (a) Phase diagram and examples of the different morphologies of the spinodal decomposition of a binary alloy with cubic elastic anisotropy, as a function of mean composition $\langle\varphi\rangle$ and misfit strain η. The reported images correspond to representative evolution stages obtained by the numerical integration of the Cahn-Hilliard equation (6.8) starting from the homogeneous phase slightly perturbed by Perlin noise around the average composition $\langle\varphi\rangle$. (b) Example of a phase-field evolution from the dataset, starting from a Perlin noise initial condition and corresponding to a misfit strain of $\eta \approx 0.48\%$.

6.3 Dataset Generation

To train the NN surrogate, a comprehensive dataset of spinodal decomposition trajectories must first be built from PF simulations. This section describes the generation of such a dataset.

Simulation Setup and Numerical Parameters

The dataset is made up of 2000 time-evolution sequences generated by integrating the PF equations (Eq. 6.7) under varied initial conditions, average compositions, and misfit strain values. All simulations were computed on a 128×128 uniform square lattice with periodic boundary conditions, naturally enforced through the spectral numerical scheme we employed (Eq. 6.8). To adequately resolve the PF boundary, the interface thickness parameter was set to $\epsilon = 5$ in units of grid spacing. The mobility coefficient M acts solely as a timescale scaling factor. For this reason, we fixed $M = 1$ and employed a time-step of $\delta t = 0.01$ to ensure numerical stability. Each simulation sequence contains 100 snapshots, sampled at regular time intervals $\tau = 100\,\delta t$, such that successive frames exhibit sufficient structural evolution to provide meaningful training information for the NN model. In the following, we express time in units of τ.

Initial Conditions and Seeding

We decided to design initial conditions corresponding to a homogeneous mixed phase with local composition fluctuations, which eventually serve as seeds for the phase-separation process, mimicking the spontaneous phase separation in cooling alloys. To generate these fluctuations, we implemented a script based on periodic Perlin noise [209], a gradient-based noise algorithm that produces spatially correlated patterns. In 2D Perlin noise, the domain is first divided into an integer lattice coarser than the pixel-pixel spacing, and each lattice point is assigned a random unit gradient vector. Periodicity is enforced by choosing the gradient vectors at one boundary of the domain to be identical to those at the opposite boundary. For any pixel of the domain, one finds the surrounding cell, computes the offset vectors from the four cell corners, and takes the dot product between each corner's gradient and its corresponding offset. These four scalar values are then interpolated using a smooth function, yielding a continuous value with continuous derivatives across cell boundaries. In this way, we can generate diverse yet quite physically plausible nucleation scenarios easily.

Parameter Space Sampling

To promote robust model generalization across a broad parameter regime, the dataset is sampled on a diverse range of initial configurations. The average composition $\langle\varphi\rangle$ was sampled from a uniform random distribution in the range $[0.2, 0.8]$. With the chosen simulation parameters, this composition range deliberately encloses the spinodal limits at zero misfit strain (approximately $\varphi \approx 0.21$ and $\varphi \approx 0.79$, as shown in Fig. 6.3), ensuring the dataset contains both phase-separated and homogeneous configurations. Each of these initial Perlin noise profiles was then multiplied by a scaling factor sampled from a normal distribution with standard deviation 0.1. In this way, by also varying the amplitude of composition fluctuations, we aim to provide additional variability and promote generalization to diverse spinodal decomposition pathways.

6.4 Convolutional Architecture for Field Evolution

In this section, we review the main neural network architectures that can be employed as surrogate models for continuum evolution problems, discussing their core strengths and limitations. In numerical simulations, any continuum field is represented on a discrete spatial grid, so the field becomes a matrix of values. This matrix can be interpreted as an image, where each pixel corresponds to the value of the discretized field at a specific location. This observation forms the basis for connecting surrogate models for continuum systems with the broader toolkit of image processing.

Surrogate modeling often begins with feature extraction from data. When the data originate from numerical simulations of discretized continuum fields, this feature extraction essentially becomes the task of identifying meaningful patterns in images and learning a mapping based on those features. A core tool for feature extraction in image processing is the convolutional filter. Convolutional filters are small matrices (typically 3×3 or 5×5 for most applications) that slide across an image and compute weighted sums of local pixel neighborhoods. By applying different filters, one can detect edges, textures, spatial patterns, or other local features that help a model understand and process image-like data.

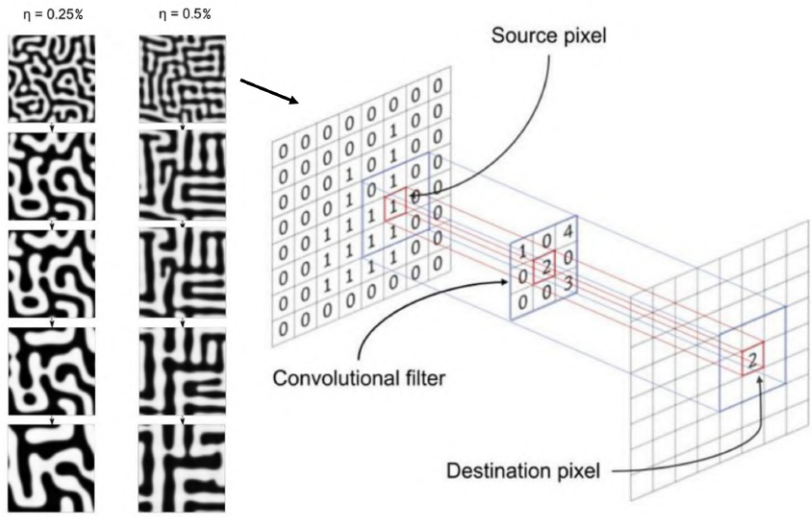

Figure 6.4: Schematic illustration of a convolutional filter applied to images of phase-field evolution. Each image is represented as a grid of scalar values. The filter slides across the image, and at each location, a convolution is performed by computing the dot product between the filter weights and the corresponding image values within the kernel window.

Simple Convolutional Neural Networks (CNNs)

Convolutional neural networks are a class of NNs that exploit learnable convolutional filters, also known as kernels, and apply them to the input field, transforming spatial patterns into feature maps. Instead of learning connections between every input and every neuron, as in a fully connected network, CNNs learn local spatially invariant connections, making them far more efficient and better suited to spatial data [210]. Moreover, multiple stacked layers are able to build hierarchical representations: early layers capture low-level features, like gradients and local composition variations, while deeper layers encode large-scale patterns, like domain structures and more complex interfacial morphologies.

In our context (accelerating the system dynamics), we want a model capable of mapping a microstructural field at time t to the field at a future time $t + \tau$:

$$\varphi(\mathbf{x}, t + \tau) = \mathrm{CNN}(\varphi(\mathbf{x}, t);\, \theta) \tag{6.10}$$

where θ represents network parameters. A common CNN architecture typically consists of multiple convolutional layers with many learned filters of customizable size $k \times k$. A simple schematic representation of

the application of a convolutional filter is shown in Fig. 6.4.

Activation functions, such as ReLU, Tanh, or others, are employed between layers to introduce nonlinearity. Optionally, one could also use tricks like pooling or striding for reducing spatial resolution (downsampling). Also, regularizing methods can be employed, such as batch normalization for training stability and dropout layers to prevent overfitting and have a built-in ensemble of models for eventual uncertainty estimation of the predictions.

CNNs are a relatively simple architecture, straightforward to train and computationally efficient, as they are naturally parallelizable on GPUs. An important aspect of fully convolutional variants (no fully connected layers) is that they preserve spatial structure and generalize to different domain sizes. From a practical standpoint, this means we can train a fully convolutional model on a dataset defined over a relatively small domain (one that is computationally manageable for a traditional PDE solver) and then apply the trained model to much larger domains, provided that locality and scale-independent dynamics assumptions hold.

6.5 Recurrent Architecture for Field Evolution

The methods discussed above face a fundamental challenge when used as surrogate models for predicting temporal evolution. While they may perform well for single-step predictions, they tend to suffer from error accumulation over multiple time steps. Even traditional numerical solvers introduce small errors at each step, which gradually compound and degrade long-time accuracy. For neural-network-based surrogate models, this issue is often more severe, as accumulated artifacts can quickly lead to highly unphysical dynamics. A direct example of this behavior will be presented in Section 7.2, where we test and compare different NN architectures.

RNNs

One way to address this issue is to incorporate not only the current state at time t but also several past states, similar to forecasting models. This approach connects surrogate modeling with the broader field of time-series analysis, which spans applications from energy demand prediction to sales forecasting, financial risk assessment, and algorithmic

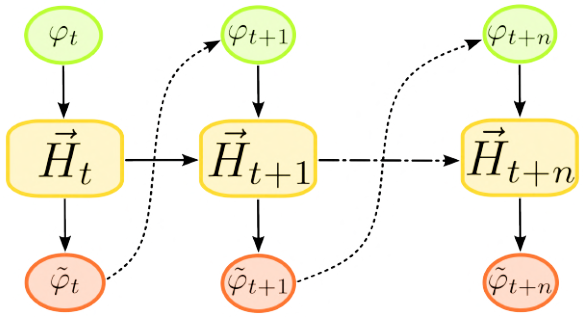

Figure 6.5: Schematic representation of a recurrent architecture. At each time step t, the current state φ_t is processed together with the hidden state $\vec{H}_t$ to produce a prediction $\tilde{\varphi}_t$ and an updated hidden state. The predicted state is then fed back into the network at the next time step, allowing the hidden state to evolve recursively and enabling the iterative generation of a sequence of predictions.

trading. A wide range of forecasting methods exists, from classical techniques such as exponential smoothing, autoregressive models, moving averages, and ARMA/ARIMA models developed in the 1980s, to more advanced approaches.

In recent years, recurrent neural network architectures have gained prominence, particularly long short-term memory networks (LSTMs) [211] and gated recurrent units (GRUs) [212]. In the following, we discuss these recurrent neural networks (RNNs) in detail and then explore how they can be combined with convolutional architectures for our purposes.

Recurrent networks maintain an internal hidden state h_t that evolves with each time step, encoding temporal context. At each time step, the hidden state is updated based on the current input and the previous hidden state:

$$h_{t+1} = f(\varphi_t, h_t; \theta) \tag{6.11}$$

and the predicted field is:

$$\tilde{\varphi}_{t+1} = g(\varphi_t; h_{t+1}) \tag{6.12}$$

where f and g are learned functions. So-called *vanilla* RNNs, depicted in Fig. 6.5, are shown to suffer from vanishing/exploding gradients during training, making it difficult to learn long-range temporal dependencies [213, 214, 215]. LSTM and GRU architectures address this by

introducing some kind of gating mechanisms that selectively preserve or forget information of the hidden states across time.

LSTMs, and GRUs

LSTMs and GRUs are special RNNs designed to keep important information around for a long time and forget irrelevant info of the system dynamics. Each LSTM cell has an internal memory represented by the cell state, and three different gates that control the flow at each temporal step:

- Forget gate: controls which info to throw away
- Input gate: controls what new info to add
- Output gate: controls what info to output.

More in detail, let φ_t be the input at time t, h_{t-1} be the previous hidden state, W, b some weights and biases, and σ the sigmoid function. At first, the forget gate filters out what to keep from the previous cell state:

$$f_t = \sigma(W_f[h_{t-1}, \varphi_t] + b_f). \tag{6.13}$$

Through the input gate, new info is selected to be added:

$$i_t = \sigma(W_i[h_{t-1}, \varphi_t] + b_i), \tag{6.14}$$

so that the memory cell state is updated as:

$$C_t = f_t \odot C_{t-1} + i_t \odot \tanh(W_c[h_{t-1}, \varphi_t] + b_c), \tag{6.15}$$

where $\odot$ represents the element-wise product. Once the cell state is updated, the hidden state output is given as:

$$h_t = \sigma(W_o[h_{t-1}, \varphi_t] + b_o) \odot \tanh(C_t). \tag{6.16}$$

GRU, on the other hand, is a simpler cousin to LSTM [212, 216]. The key difference here is that GRU merges the forget and input gates into a single update gate. In this way, GRUs have only two gates:

- Update gate: controls whether to keep past info or use the new candidate
- Reset gate: helps create the candidate by forgetting parts of the old hidden state.

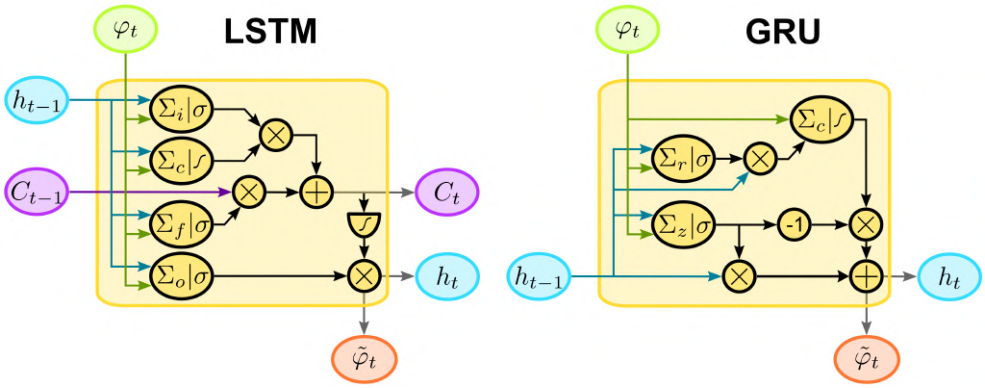

Figure 6.6: Schematic representation of the long short-term memory (LSTM) and gated recurrent unit (GRU) architectures.

This simpler structure often trains faster, still with similar performance. More in detail, in the update gate, the quantity

$$z_t = \sigma(W_z[h_{t-1}, \varphi_t] + b_z) \tag{6.17}$$

is computed, controlling how much of the hidden state should be updated; while in the reset gate, the quantity

$$r_t = \sigma(W_r[h_{t-1}, \varphi_t] + b_r) \tag{6.18}$$

is computed, controlling how much of the previous hidden state to forget before computing the new candidate. At the final step, the new hidden state is computed:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tanh(W_h[r_t \odot h_{t-1}, \varphi_t] + b_h). \tag{6.19}$$

These recurrent architectures, thanks to their hidden states, can explicitly capture temporal correlations, reducing error accumulation over multiple steps. Moreover, they offer quite a lot of flexibility, since they can handle variable-length input sequences and predict sequences of arbitrary length as well. A schematic representation of both the LSTM and GRU architectures is shown in Fig. 6.6

6.6 Other novel approaches

U-Net and Encoder-Decoder Architectures

A key limitation of simple CNNs is their dependence on the resolution and locality imposed by their receptive fields. If the system exhibits long-range interactions that extend beyond the kernel's receptive field, a standard CNN will struggle to capture these effects. One way to mitigate this is to use larger kernels or, more effectively, to stack additional convolutional layers to increase the effective receptive field. An even more powerful approach is to adopt a U-Net architecture, which is designed to better capture multi-scale and long-range features.

The core concept of U-Nets is to employ an encoder-decoder architecture with a bottleneck structure to capture multi-scale features [217, 218]. This architecture is composed of four main parts:

- **Encoder**: Series of convolutional and pooling layers that downsample the input, progressively extracting coarse features
- **Bottleneck**: Deepest layer with the smallest spatial resolution but highest feature density
- **Decoder**: Series of upsampling and convolutional layers that progressively restore spatial resolution
- **Skip connections**: Direct links between layers in the encoder directly to the corresponding layers in the decoder at the same spatial resolution.

Downsampling through pooling and repeated convolutions is the key to enlarging the receptive field in a computationally efficient way, but at a cost. These operations in the encoder lead to a loss in spatial resolution, and the information flowing through the bottleneck is thus limited, especially for small details. Moreover, without skip connections, the decoder would rely only on compressed, abstract features from the bottleneck, and fine spatial information and boundaries would likely not be recovered, leading to blurry, inaccurate segmentation of small structures. Without the skip connections, a U-Net would be just an autoencoder. The skip connections are thus critical: they allow the network to use both coarse contextual information (from the bottleneck) and fine spatial details (from early encoder layers) when generating the output.

More in detail, in a skip connection, the feature map output from an encoder layer before spatial downsampling (e.g., after convolutions,

before pooling) is copied and then concatenated with the feature map produced by the decoder's upsampling layer at the matching resolution. With skip connections, all information in the channels from both the encoder and decoder at that stage is preserved. By doing so, the network can leverage both low and high-level features during upsampling in the decoder part.

Another positive impact of skip connections is the improvement in gradient flow: they act as alternative paths for gradients, alleviating vanishing gradients in deep networks and making the network easier to train, in a similar way to ResNet architectures.

Deep Operator Networks (DeepONets)

Neural networks are well known to be universal approximators of continuous functions. Less recognized until recently is the result that NNs can also approximate nonlinear continuous operators [219]. Deep Operator Networks (DeepONets) [220] provide a practical framework for learning such operators. A DeepONet architecture consists of two subnetworks: a *branch network*, which takes a discretized representation of the input function $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_n)$, and a *trunk network*, which receives the coordinates at which the output function $u = \mathcal{G}[\nu]$ is evaluated. This architecture enables DeepONets to learn a wide class of operators, including explicit operators such as integrals and fractional Laplacians, as well as implicit operators corresponding to deterministic and stochastic differential equations. As a result, DeepONets have been successfully applied across diverse domains, including phase-field modeling [221, 15].

Fourier Neural Operators (FNOs)

Fourier Neural Operators (FNOs) [222] formulate operator learning in Fourier space, leveraging the decomposition of physical fields into frequency modes. Low-frequency components capture smooth, global dynamics, such as domain coarsening in phase-field models, while higher frequencies represent fine interfacial structures. FNOs learn a parametric kernel applied via spectral convolutions, allowing efficient propagation of features and excellent scalability to high-resolution grids without retraining. Their main strengths are translation equivariance and computational efficiency. However, they struggle with sharp discontinuities or non-periodic boundary conditions. Despite their recent introduction, FNOs have already demonstrated substantial acceleration of phase-field simulations while maintaining high accuracy [223].

To address some of these limitations, hybrid architectures such as U-AFNO combine U-Net-style multiscale feature extraction with adaptive Fourier operator blocks. This approach balances local spatial details with global features and has proven particularly effective for chaotic and multiphase systems where pure FNOs may blur interfaces [224].

Future architectural explorations

Transformer architectures, originally developed for natural language processing, rely on self-attention mechanisms to capture long-range dependencies [225]. Recently, they have been adapted to operator learning by treating spatial grids as sequences augmented with positional encodings that preserve geometric structure. Attention mechanisms can enable information exchange between distant regions of the domain, in principle allowing transformers to capture collective phenomena such as long-range coarsening dynamics that are difficult for purely local convolutional models.

A notable recent development is the Physics-Informed Transformer Neural Operator (PINTO) [226], which addresses key challenges in solving nonlinear PDEs. PINTO learns mappings between function spaces using only PDE residuals, without requiring simulation data or retraining for new initial or boundary conditions.

Finally, diffusion-based and flow-based generative models represent an emerging direction in sequence generation. While they have not yet been widely applied to accelerating phase-field simulations, their ability to model complex stochastic dynamics suggests significant potential for future developments where one-to-one fidelity is not required. In particular, in Ref. [227], rolling diffusion models have been introduced, showing great forecasting capabilities in predicting complex temporal dynamics, such as in the case of chaotic fluid dynamics forecasting experiment.

Overall, the methods discussed here are not exhaustive of the rapidly expanding landscape of approaches, still they highlight promising directions for next-generation and physics-aware surrogate modeling.

Chapter 7

Deep Learning For Coherent Spinodal Decomposition

Building on the theoretical and methodological foundations established in previous sections, this section demonstrates the practical application of artificial neural network architectures as surrogate models for accelerating continuum-scale materials simulations. Specifically, as a case study, we want to accelerate PF simulations of spinodal decomposition in coherently strained alloy systems. Extending PF simulations to large computational domains or significantly long timescales can become computationally expensive, restricting the exploration of complex material behaviors.

We present a deep learning framework that learns to replicate the complex dynamics of spinodal decomposition in cubic alloy microstructures with anisotropic elastic constants. The surrogate model is trained on the dataset discussed in Section 6.3, comprising PF simulations spanning a wide range of composition fields and misfit strain parameters. As we already saw in Chapter 6, these parameters control phase morphology and kinetics in these systems. By explicitly conditioning the NN on the misfit parameter, we enable the model to capture how elastic anisotropy governs the evolution of microstructural features.

In this chapter, we also extensively test the surrogate model's predictive performance through multiple analyses. First, we establish accuracy benchmarks through pixel-level, one-to-one comparisons with ground-truth PF simulations within the training data regime. Second, we rigorously test generalization capabilities across longer timescales, larger spatial domains, and different parametric conditions. Third, we examine whether the model implicitly learns the underlying physical princi-

ples governing spinodal decomposition, assessed through its ability to predict global thermodynamic quantities and reconstruct phase stability diagrams.

After performing such validation across multiple performance metrics, we demonstrate that NN-based surrogates can effectively accelerate continuum materials simulations while preserving physical fidelity, enabling new studies at larger scales at a fraction of the cost compared to standard approaches.

7.1 Methods: CRNN Architecture

A description of the convolutional and recurrent neural network architectures was already done in Chapter 6. In this section, we provide a detailed discussion of the specific framework we developed by combining these two architectures into a convolutional recurrent NN (CRNN). We discuss the hyperparameters employed, along with the physics-inspired design of the output layer and the regularized loss function we implemented to enable more effective training.

The model mainly builds on top of previous CRNN designs developed in the research group and showcased in Ref. [228] and Ref. [20]. A key difference is the explicit incorporation of parametric conditioning on the misfit strain, enabling the network to predict noticeably different morphological field evolution across a continuous range of elastic conditions.

Model Objective and Input Structure

The objective of our CRNN model is to predict the temporal evolution of the composition field φ from a given initial configuration φ_0 and a specified value of the misfit parameter η, which is supplied. At each time step, the network receives the composition field φ_t as a spatial map. To condition the evolution on the misfit parameter, the composition field φ_t is concatenated with η at every time step, forming a two-channel input representation: one channel containing the spatially varying composition and the other a uniform field equal to η. This straightforward approach offers practical advantages: it is simple to implement, readily extensible to multiple parameters or even multiple spatially varying fields, and preserves the fully convolutional nature of the architecture, which is essential for generalization to arbitrary domain sizes [228].

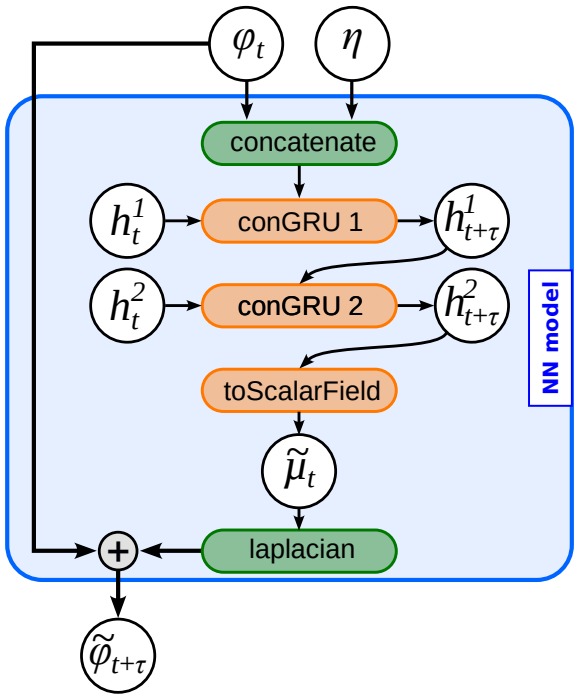

Figure 7.1: Schematic illustration of the convolutional recurrent neural network (CRNN) used for evolution prediction. At each time step, the model advances the system from the current state φ_t to the predicted state $\widetilde{\varphi}_{t+\tau}$, using the hidden states h_t^i (with i indexing the convolutional GRU submodules) and a control parameter η supplied as a constant field. Instead of predicting $\widetilde{\varphi}_{t+\tau}$ directly, the network outputs an intermediate scalar field $\widetilde{\mu}_t$, from which the updated state is subsequently computed. Green boxes denote fixed operations, while orange boxes indicate learnable components.

Core Architecture: Stacked Convolutional GRU Blocks

The input data are processed through two stacked convolutional gated recurrent unit (convGRU) blocks [212]. Each convGRU block has 24 channels and employs 5×5 convolutional kernels. These architectural choices were determined through preliminary hyperparameter tuning to optimize, in a balanced way, both model performance and training stability. This configuration yields approximately $150,000$ trainable parameters, ensuring manageable computational demands during both training and inference. Periodic boundary conditions are enforced through circular padding, maintaining consistency with the PF simulation dataset [229].

Physics-Inspired Output Layer

Rather than directly predicting the composition field at the next time step $\widetilde{\varphi}_{t+\tau}$, the network outputs an auxiliary scalar field $\widetilde{\mu}_t$, which represents the local chemical potential (scaled by mobility M) as it appears in the Cahn-Hilliard equation (6.7). The updated composition field is then computed as:

$$\widetilde{\varphi}_{t+\tau} = \widetilde{\varphi}_t + \nabla^2 \widetilde{\mu}_t \tag{7.1}$$

This design choice introduces a strong physics-based regularization by construction: the network cannot produce non-conservative dynamics, as mass conservation is mathematically enforced by the Laplacian operation. This constraint significantly improves model stability and reduces error accumulation during temporal extrapolation beyond the training regime.

A schematic representation of the workflow of our architecture is shown in Fig. 7.1

Loss Function and Regularization

The model is trained using a composite loss function that combines three components, each with its own weight:

$$\mathcal{L}(\theta) = \frac{1}{N_{\text{ts}} T} \sum_{i=1}^{N_{\text{ts}}} \sum_{t=1}^{T} \left(\mathcal{L}_\varphi(\theta) + \lambda_\nabla \mathcal{L}_\nabla(\theta) + \lambda_f \mathcal{L}_f(\theta) \right). \tag{7.2}$$

The three loss terms are:

- $\mathcal{L}_\varphi$: Mean squared error between predicted and ground-truth composition fields, measuring pixel-level accuracy:

$$\mathcal{L}_\varphi(\theta) = \langle (\varphi_i(t) - \widetilde{\varphi}_i(t|\theta))^2 \rangle \tag{7.3}$$

- $\mathcal{L}_\nabla$: Mean squared error between the squared gradients of predicted and true fields:

$$\mathcal{L}_\nabla(\theta) = \langle \left(|\nabla \varphi_i(t)|^2 - |\nabla \widetilde{\varphi}_i(t|\theta)|^2 \right)^2 \rangle \tag{7.4}$$

- $\mathcal{L}_f$: Mean squared error between the free energy densities of predicted and true configurations:

$$\mathcal{L}_f(\theta) = \langle f(\varphi_i(t)) - f(\widetilde{\varphi}_i(t|\theta)) \rangle^2. \tag{7.5}$$

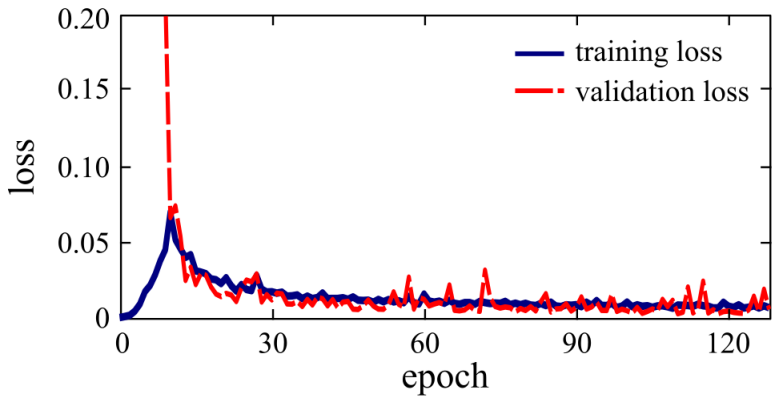

Figure 7.2: Training and validation loss curves during training; the pronounced fluctuations in the first 10 epochs arise from the curriculum learning strategy employed.

Here, θ denotes the network parameters, i indexes the N_{ts} elements of the training sequences, and t represents the time step ranging from 1 to the total sequence length T. The symbol $\langle . \rangle$ represents spatial average.

The weights $\lambda_\nabla = 60$ and $\lambda_f = 150$ were selected based on preliminary analysis to balance the relative importance of these regularization terms. Notice that here we compute the mean free-energy density $\langle f \rangle \approx 2\langle w \rangle + \langle \rho \rangle$ using the approximation $\varepsilon |\nabla \varphi|^2 \approx w(\varphi)$ (see e.g. Ref. [230]). The gradient-based term $\mathcal{L}_\nabla$ suppresses unphysical local fluctuations, while the free-energy term $\mathcal{L}_f$ ensures that predictions remain thermodynamically consistent even when pixel-level accuracy degrades.

In Sect. 7.3, we highlight the importance of incorporating multiple regularization terms in the loss function by comparing our model with an alternative trained using a simpler objective that accounts only for the mean squared error between the true and predicted fields φ.

7.2 Training, Validation, and Testing

Having established the dataset characteristics and comprehensively described the model architecture, we now present the performance assessment of the trained surrogate model. This section evaluates model performance during training and validation, examines accuracy on an independent test set, and establishes the regimes in which the model maintains satisfactory predictive fidelity.

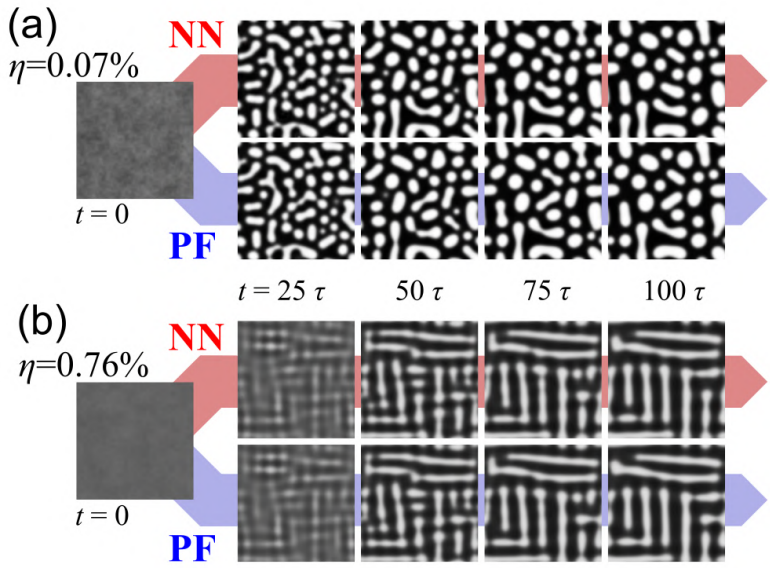

Figure 7.3: Comparison between NN predictions and ground-truth phase-field (PF) evolutions for two validation examples, corresponding to (a) low ($\eta \approx 0.07\%$) and (b) high ($\eta \approx 0.76\%$) misfit, respectively.

Training Strategy and Data Augmentation

The full dataset of 2000 simulations was divided with a $4:1$ ratio between training and validation subsets. Data augmentation techniques, such as spatial reflections, 90° rotations, and the symmetry transformation $\varphi \to 1 - \varphi$, were applied to increase dataset diversity by leveraging the physical symmetries of the system, even though these symmetries are not explicitly built into the model architecture. Additionally, noise injection regularization was employed during training by adding small Gaussian perturbations (with standard deviation $1 \cdot 10^{-3}$) to the composition fields, improving robustness to input variations.

Training was performed using the Adam optimizer [53] with a curriculum learning strategy [231]. In this approach, the model initially predicts only the final time step while being provided with nearly complete sequences (99 snapshots at first). As training progresses, the number of input frames is gradually reduced, requiring the model to predict increasingly longer portions of each sequence. By the final stage of training, the model receives only the initial condition and must predict the full 100-step sequence iteratively. While this procedure can temporarily increase the loss during this so-called ramp length, as shown in Fig. 7.2, it facilitates more stable learning by progressively increasing task com-

plexity. Once stabilized, loss fluctuations remain modest, consistent with expected behavior in stochastic optimization.

Validation Set Performance

The model exhibiting the lowest validation loss was retained for all subsequent analyses. To confirm that the trained model accurately captures the dynamics of spinodal decomposition, visual comparisons were performed on representative validation sequences. Fig. 7.3 shows predicted and ground-truth evolutions across the parametric range: a low-misfit case ($\eta \approx 0.07\%$, $\langle\varphi\rangle \approx 0.396$) and a high-misfit case ($\eta \approx 0.76\%$, $\langle\varphi\rangle \approx 0.388$). Both cases demonstrate nearly one-to-one correspondence between predicted and reference sequences, with only minor discrepancies in domain connectivity. This visual agreement suggests that the model has successfully learned the essential features of spinodal decomposition across the parametric space.

Test Set Design and Temporal Generalization

To rigorously assess robustness and generalization capability, an independent test set of 500 sequences was generated with identical characteristics to the training data but extended to $500\,\tau$: five times longer than the training sequences. This design enables evaluation of the model's ability to predict dynamics extrapolating beyond the training regime. In Fig. 7.4, we show an example taken from the test set ($\eta \approx 0.45\%$) showing the predictions over extended times. Within the training time window $[0, 100]\,\tau$, the model maintains the near-perfect one-to-one match observed on validation data. However, as evolution progresses beyond this range, pixel-level accuracy gradually degrades. This degradation stems from the progressive accumulation of small prediction errors compounded by critical dynamical events such as domain coalescence and splitting. At later stages ($t = 500\,\tau$), the predicted morphology remains physically plausible but diverges in one-to-one accuracy from the ground-truth simulation, as illustrated by the error map $|\Delta\varphi| = |\varphi_{pred} - \varphi_{true}|$ in Fig. 7.4.

Quantitative Error Analysis: Pixel-Level Metrics

To quantify prediction accuracy, the mean absolute error (MAE) was computed at each time step. The statistical distribution of MAE across the test set, and as a function of time, reveals sublinear error accumulation, remaining as low as ≈ 0.025 within the full training window

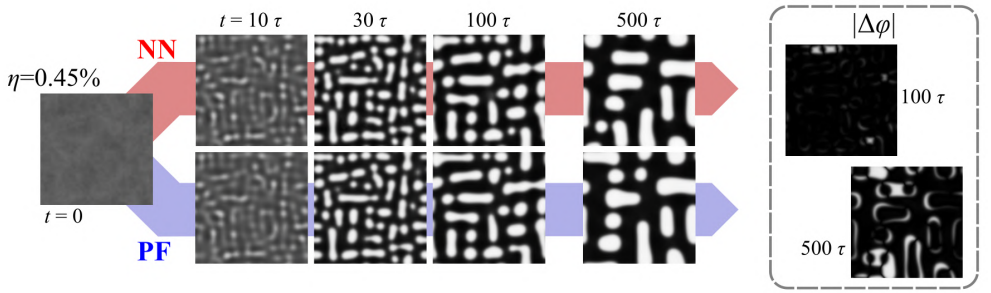

Figure 7.4: Test sequence of spinodal decomposition at misfit $\eta \approx 0.45\%$, comparing the NN prediction with the reference phase-field (PF) evolution, initialized from a Perlin noise profile with $\langle\varphi\rangle = 0.5$. Maps of the absolute error, defined as the pointwise difference $|\Delta\varphi| = |\varphi - \widetilde{\varphi}|$, are shown at $t = 100, \tau$ and $500, \tau$.

$[0, 100]\,\tau$ and reaching a median of approximately 0.09 when extrapolating to $500\,\tau$, as shown in Fig. 7.5a. These error magnitudes align with typical extrapolation errors reported in comparable NN surrogate studies [15, 16] and reflect the inherent sensitivity of time-evolving systems to small perturbations. Notably, even PF simulations exhibit comparable sensitivity to numerical discretization choices, demonstrating that the observed deviations are not unique to the NN model predictions but rather are characteristic of bifurcation-sensitive dynamical systems like the one simulated here.

Beyond Pixel-Level Accuracy: Global Descriptors

While pixel-level metrics reveal accuracy degradation during extrapolation, assessing physical fidelity also requires evaluation of global, morphology dependent quantities. The total interface length, normalized by domain size as $2\langle|\nabla\varphi|\rangle$, provides such a global descriptor. Fig. 7.5b demonstrates that the relative error in interface length prediction grows substantially more slowly than pixel-level MAE, remaining below 2% throughout the training window and under approximately 5% while extrapolating at $500\,\tau$ for the majority of cases, as reflected in the interquartile range. This significant discrepancy between local and global error metrics indicates that while the model loses one-to-one pixel correspondence during extrapolation, it continues to provide physically reasonable representations of overall domain evolution and morphological statistics.

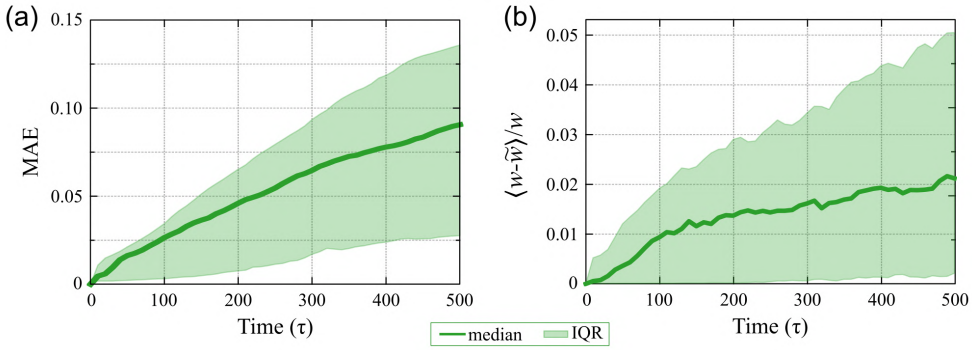

Figure 7.5: (a) Time evolution of the mean absolute error (MAE) and (b) of the relative error in interface length, computed as $\langle w(\varphi) \rangle$, between NN predictions and PF ground truth, for a test set of 500 sequences up to $500, \tau$. The solid line denotes the median, while the shaded area indicates the interquartile range (IQR).

7.3 Time Evolution of Mean Properties

While the previous section focused on pixel-level accuracy, this section extends the analysis to assess the physically meaningfulness of the model's predictions at longer timescales. Although the surrogate model loses one-to-one correspondence with ground-truth simulations during temporal extrapolation, we verify whether it continues to satisfy the underlying thermodynamic principles governing spinodal decomposition.

Thermodynamic Consistency: Free Energy Evolution

Since the Cahn-Hilliard equation inherently drives the system toward free energy minimization, the total free energy density $\langle f \rangle$ serves as a primary check of thermodynamic fidelity. To evaluate this, a new dedicated test set of 50 independent evolutions was built for fixed average composition $\langle \varphi \rangle = 0.5$ and misfit strain $\eta = 0.6\%$, each extending to $1000\,\tau$: ten times longer than the training sequences. Fig. 7.6a presents the temporal evolution of $\langle f \rangle$, with solid lines indicating ensemble medians and shaded regions denoting the interquartile range (IQR).

The NN predictions demonstrate excellent agreement with PF ground truth even in the extrapolation regime, with median values of the two approaches falling within each other's IQR for most of the evolution. Deviations become apparent only at later snapshots (approximately $900\,\tau$). Importantly, this threshold is substantially further in time than the point where pixel-level errors become evident, demonstrating that global ther-

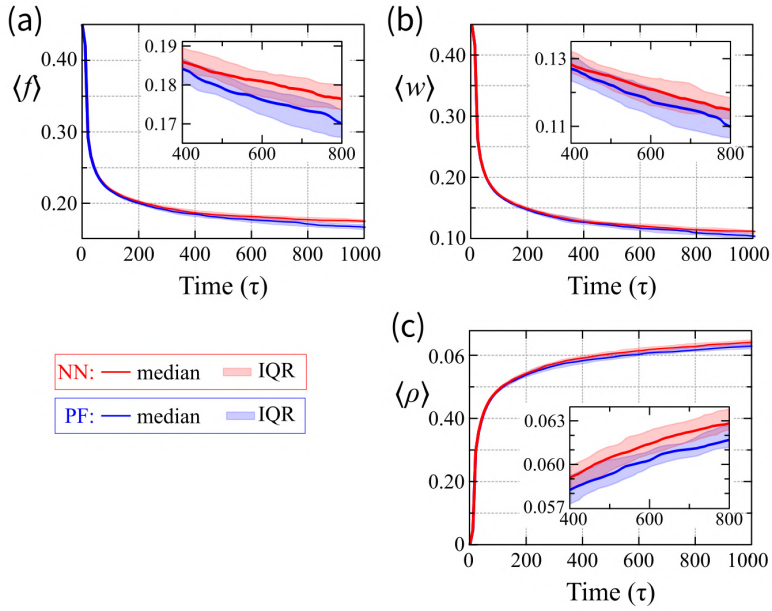

Figure 7.6: Time evolution of (a) the mean free energy density $\langle f \rangle$, (b) the mean interface energy density $\langle w \rangle$, (c) the mean elastic energy density $\langle \rho \rangle$.

modynamic behavior still persists beyond the point of one-to-one correspondence.

Energy Decomposition: Interface and Elastic Contributions

A more detailed analysis reveals how the surrogate captures the competing energy contributions governing the dynamics. Fig. 7.6b,c separately display the mean interface energy density $\propto 2\langle|\nabla\varphi|\rangle$ and mean elastic energy density $\langle \rho \rangle$. While the CRNN model exhibits a slight underestimation of interface energy decay and a slight overestimation of elastic energy growth, both components remain consistent with PF results within the IQR. Notably, the relative errors in the two contributions are comparable in magnitude, suggesting the model is capable of preserving a balance between interfacial and elastic effects in the domain morphology evolution.

Training with an unregularized MSE loss

In Sect. 7.1, we introduced the composite loss function adopted in this work, which combines multiple terms: the MSE between the predicted ($\widetilde{\varphi}_t$) and ground-truth (φ_t) composition fields, the MSE between their

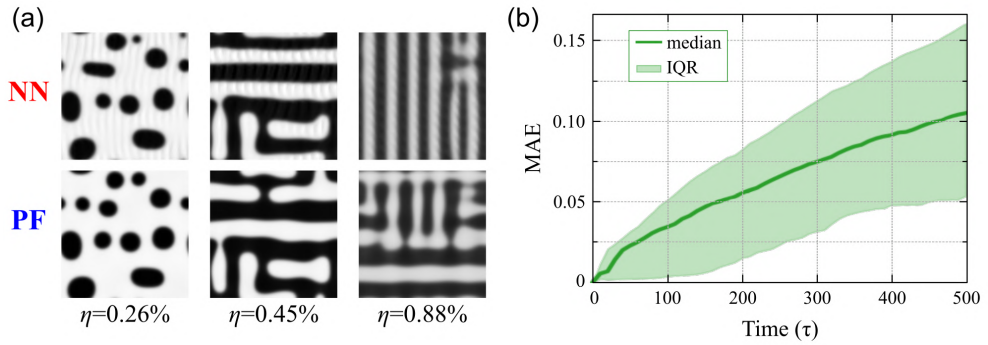

Figure 7.7: (a) Comparison between configurations obtained with a model trained without loss regularizations and the corresponding ground truth. Simulations are taken from the test set used for Fig. 7.5 at time $t = 500\,\tau$ for three different η misfit values. (b) Time evolution of MAE computed on the same test set. The median value is drawn by a solid line while the shaded area indicates the IQR.

squared gradients, and the MSE between the corresponding free energy densities. Here, we explicitly demonstrate the importance of including these additional regularization terms.

To this end, we train an alternative model using a simpler loss function defined solely by the MSE on the composition field:

$$\mathcal{L}(\theta) = \frac{1}{N_{\text{ts}}T}\sum_{i=1}^{N_{\text{ts}}}\sum_{t=1}^{T}\langle(\varphi_i(t) - \widetilde{\varphi}_i(t|\theta))^2\rangle. \tag{7.6}$$

Using this unregularized objective, we repeat the same set of tests performed for the main model and directly compare the results. We first consider the test set of 500 sequences of length $500, \tau$ used previously. As shown in Fig. 7.7a, which reports PF and NN-predicted snapshots at $t = 500, \tau$, visual inspection already reveals the presence of spurious undulations in the $\varphi \approx 1$ (white) phase. These artifacts are absent in the regularized model, primarily due to the strong smoothing effect introduced by the gradient-based loss term $\mathcal{L}_\nabla$.

We also evaluate the time evolution of the MAE, shown in Fig. 7.7b. The local prediction error of the unregularized NN remains relatively small and only slightly exceeds that of the fully regularized model, indicating that the additional loss terms have a limited impact on pixel-level error accumulation.

However, more pronounced differences emerge when considering physically meaningful observables. We compute the time evolution of the

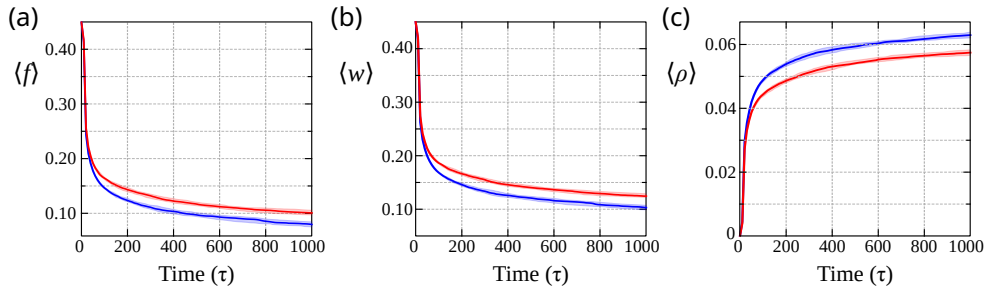

Figure 7.8: Time evolution of (a) mean free energy density $\langle f \rangle$, (b) mean interface energy density $\langle w \rangle$ and (c) mean elastic energy density $\langle \rho \rangle$ computed on the same test set of 50 simulations used for Fig. 7.6 (with fixed $\langle w \rangle = 0.5$ and $\eta = 0.6\%$) by using the NN model trained without loss regularizations (red) and the PF ground truth solutions (blue).

mean free energy density $\langle f \rangle$, the mean interface energy density $\langle w \rangle$, and the mean elastic energy density $\langle \rho \rangle$ using the same test set as in Fig. 7.6. As shown in Fig. 7.8, discrepancies between the PF reference trajectories and those predicted by the NN trained without physics-inspired regularization appear already at short times and grow rapidly, ultimately rendering the predictions unreliable. This comparison clearly demonstrates that the inclusion of the additional physics-informed loss terms, $\mathcal{L}_\nabla$ and $\mathcal{L}_w$, is crucial for improving generalization and preserving the physical fidelity of the learned dynamics.

Geometric Descriptors

Beyond energetic quantities, we provide additional validation of the model's ability to capture geometrical properties. Two metrics sensitive to domain geometry were evaluated:

Isotropy Degree (I): This metric quantifies the directional balance of domain boundaries through analysis of histograms of oriented gradients computed on each snapshot [232]. Values close to unity indicate nearly isotropic patterns, while values near zero reflect preferential alignment of interfaces along crystallographic directions ($\langle 10 \rangle$ and $\langle 01 \rangle$). The CRNN reproduces this quantity accurately despite it not being included in the training loss function, demonstrating implicit learning of elastic anisotropy effects, as shown in Fig. 7.9a. More in detail, each image is partitioned into local cells of 4×4 pixels, which are then combined into 2×2 blocks to perform local normalization. Gradient orientations are discretized into 10 angular bins spanning the range $0°$ to $180°$. After

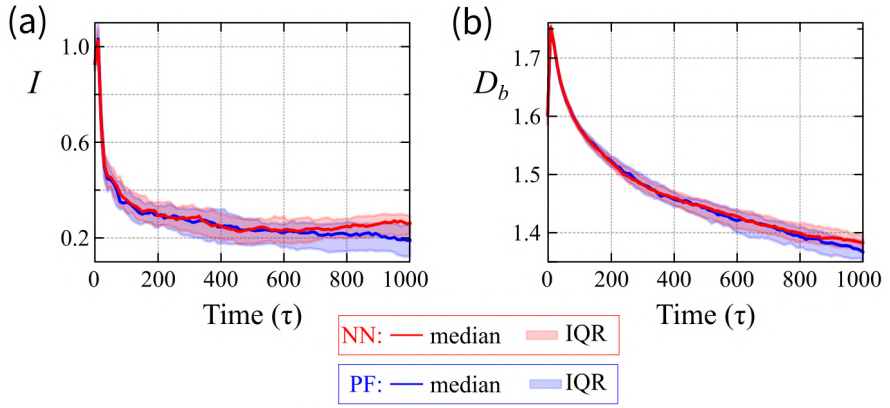

Figure 7.9: (a) The isotropy factor I, and (b) the boundary fractal dimension D_b, obtained from 50 independent simulations with fixed $\langle\varphi\rangle = 0.5$ and $\eta = 0.6\%$, and different initial Perlin noise profiles. Solid lines indicate the median, while shaded regions denote the IQR. NN predictions are shown in red, and the corresponding PF reference results are in blue.

spatial averaging, the mean gradient magnitude within each orientation bin provides a measure of the prevalence of interfaces aligned along that direction. The isotropy factor I is then defined as the ratio between the average magnitude associated with non-axial orientations, $\langle M_{\langle\neg 10\rangle}\rangle$, and that corresponding to the horizontal and vertical $\langle 10\rangle$ directions, $\langle M_{\langle 10\rangle}\rangle$:

$$I = \frac{\langle M_{\langle\neg 10\rangle}\rangle}{\langle M_{\langle 10\rangle}\rangle}. \tag{7.7}$$

Boundary Fractal Dimension (D_b): This descriptor characterizes the geometrical complexity of the domain interfaces [233]. It is obtained by assessing how the number of edge pixels scales with the observation length scale using a box-counting approach. Specifically, D_b is computed as the negative slope of the log–log relationship between the number of occupied boxes $N(L)$ and the box size L:

$$D_b = -\frac{d\log N(L)}{d\log L}. \tag{7.8}$$

The intuitive picture is that higher values indicate more irregular and convoluted boundaries. As shown in Fig. 7.9b, the model predictions closely match the ground-truth evolution of this metric, further confirming, once again, the preservation of morphological detail beyond pixel-level correspondence.

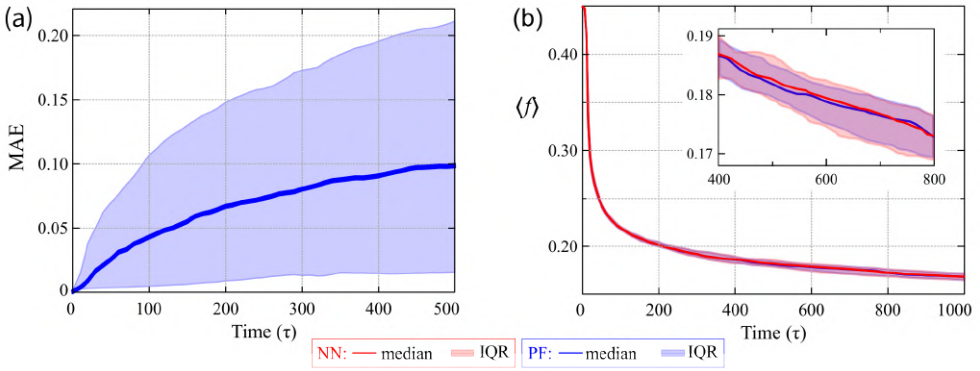

Figure 7.10: (a) Time evolution of the MAE between noisy PF simulations and the corresponding deterministic ground-truth trajectories from the test set in Fig. 7.5, defined as $\mathrm{MAE} = \langle |\varphi_{\mathrm{noise}} - \varphi| \rangle$. (b) Time evolution of the mean free energy density $\langle f \rangle$, comparing noisy PF simulations with the corresponding NN predictions for the test set shown in Fig. 7.6. Solid lines indicate median values, while shaded regions represent the IQR. Noisy PF results are shown in blue, and NN predicted dynamics in red.

Phase field simulations with noise

To contextualize the magnitude of the prediction error, we compare it with the effects of stochastic perturbations in the PF dynamics. To this end, the integration scheme of Eq. (6.8) was modified by adding Gaussian random noise ν to φ_t at each time step:

$$\widehat{\varphi}_{t+\delta t} = \frac{\widehat{\varphi_t + \nu} - \delta t M q^2 w'(\widehat{\varphi_t + \nu})}{1 + \delta t M \left(\epsilon q^4 + \eta^2 \mathrm{tr}(\mathbf{B}) q^2\right)} \tag{7.9}$$

Exact mass conservation was enforced by subtracting the spatial mean of the noise over all pixels.

For a quantitative analysis, we considered as ground truth the same 500 PF sequences used in Fig. 7.5. For each sequence, an additional PF simulation was performed starting from the identical initial condition but including stochastic noise. Fig. 7.10a reports the mean absolute error (MAE), computed over these sequences as $\mathrm{MAE} = \langle |\varphi_{\mathrm{noise}} - \varphi| \rangle$. As expected, the MAE increases over time due to the cumulative effect of random perturbations, and it would also grow with the noise standard deviation. By comparing this trend with the one in Fig. 7.5a, we find that a noise amplitude with standard deviation 7.5×10^{-4} produces a variability comparable to the prediction error of the trained NN model.

Notably, the IQR of the noisy PF ensemble is significantly wider, indicating that the NN exhibits a more concentrated error distribution for a similar median value.

As already discussed, the MAE only captures pixel-wise discrepancies and does not reflect agreement in more coarse-grained physical observables, such as free-energy evolution. To enable a direct comparison with the NN analysis in Fig. 7.6, we therefore examined the same set of 50 independent evolutions with $\langle\varphi\rangle = 0.5$ and $\eta = 0.6\%$, comparing them with the corresponding noisy PF dynamics. Fig. 7.10b shows the comparison between the time evolution of the mean free-energy density $\langle f\rangle$ of the noisy PF simulations and the NN predictions that were already reported in Fig. 7.6a. The close agreement in both median values and interquartile ranges demonstrates that the accumulation of NN prediction errors is effectively equivalent to introducing random fluctuations into the PF dynamics. Since such fluctuations do not alter the ensemble thermodynamic behavior, this comparison supports the physical consistency of the NN-generated trajectories, even when strict pointwise correspondence with the ground truth is lost due to bifurcations in topological events, such as domain pinching or merging.

Implications for Extended Extrapolation

Collectively, these findings establish that the neural network surrogate produces physically meaningful predictions for time scales extending 8–9 times beyond the training regime. While morphological detail inevitably deteriorates, the underlying thermodynamic principles and geometric properties of the system remain well-represented, enabling the model to serve as a reliable accelerator for parametric studies and long-timescale investigations that prioritize global system behavior over pixel-level fidelity.

7.4 Domain-Size Generalization Tests

Having established the model's temporal generalization capabilities, we now examine its ability to handle spatial domains substantially larger than those encountered during training. A key architectural advantage of the fully convolutional design is its inherent independence from domain dimensions: once trained, the model can in principle operate on arbitrarily sized computational grids. In this section, we indeed test the model's accuracy when applied to significantly larger domains.

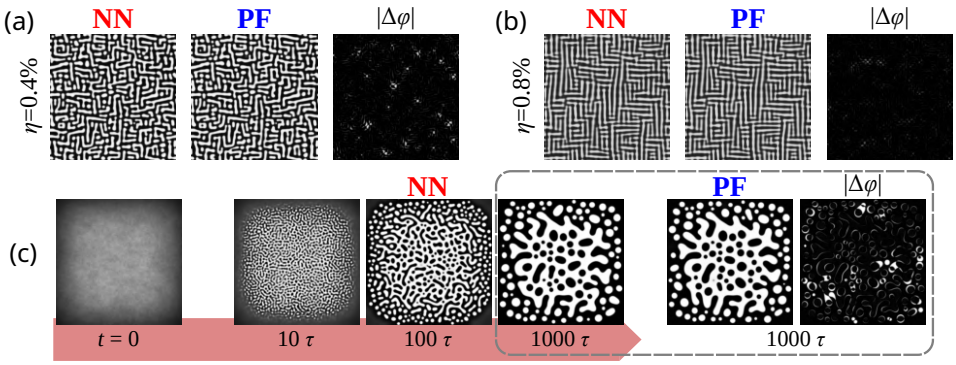

Figure 7.11: (a,b) Comparison between NN-predicted and PF reference profiles for spinodal decomposition at $t = 100, \tau$, starting from Perlin noise initial conditions on large 512×512 domains, for (a) $\eta = 0.4\%$ and (b) $\eta = 0.8\%$. The corresponding maps of absolute error over the domain are also shown. (c) Representative $1000, \tau$-long evolution on a 512×512 domain with $\eta = 0\%$, initialized from a spatially varying composition field with higher concentration at the center and lower values near the boundaries.

Large-Domain Performance: 512×512 **Simulations**

To assess spatial generalization, the trained model was tested on 512×512 computational domains, 16 times larger in area than the 128×128 training grids. In Fig. 7.11a,b, we show comparisons between NN predictions and PF ground truth for two representative cases with average composition $\langle \varphi \rangle = 0.45$ and misfit strains $\eta = 0.4\%$ and $\eta = 0.8\%$, evolved up to $t = 100\,\tau$. Despite the increase in domain size, the predicted morphologies exhibit excellent agreement with the true sequences, accurately reproducing the dominant structural features that emerge during the evolution. Again, minor local discrepancies appear where domain pinching or coalescence events occur, reflecting the same sensitivity to bifurcations observed in previous tests.

Combined Generalization: Non-Standard Initial Conditions

As a more stringent test of model robustness, we evaluated performance on initial conditions that deviate qualitatively from the training distribution while simultaneously challenging both spatial and temporal generalization. In Fig. 7.11c, we show a case initialized with a spatially inhomogeneous average composition: higher concentration in the domain center and progressively lower toward the edges. This configuration is

fundamentally different from the homogeneous Perlin noise patterns used during training. The simulation was performed on a 512×512 grid and extended to $1000\,\tau$, combining domain-size scaling ($16\times$ larger), temporal extrapolation ($10\times$ longer), and distributional variation in initial conditions. Despite these compounded challenges, the CRNN prediction still remains in substantial accordance with the PF ground truth throughout the evolution. While pixel-level differences accumulate at later times, consistent with the behavior observed in Sect. 7.2, the model successfully captures the overall morphological evolution and the spatial heterogeneity of phase separation driven by the composition gradient. This demonstrates that the learned dynamics are not narrowly specialized to the specific noise patterns of the training set but instead reflect a more general "understanding" of the underlying physics.

7.5 NN Reconstruction of the Phase Diagram

We now assess the capability of the CRNN model to extract physically meaningful insights from parametric studies. Specifically, we evaluate whether the model can accurately reconstruct the phase diagram of the system, delineating regions of phase stability as a function of composition and misfit strain.

Phase diagram reconstruction

The phase diagram reconstruction test utilizes the same test set of 500 sequences described in Sect. 7.2, which samples $(\langle\varphi\rangle, \eta)$ parameter pairs with mean composition in the $[0.20, 0.80]$ range and misfit in the $[0.0\%, 1.2\%]$ range. For each simulation, we classify the final state at $t = 100\,\tau$ as either phase-separated or homogeneous by evaluating the compositional variation $\Delta\varphi = \max(\varphi) - \min(\varphi)$. If this quantity falls below an arbitrarily chosen threshold of 0.05, the system is classified as remaining in a single homogeneous phase; otherwise, it is considered to have undergone phase separation.

In Fig. 7.12a, we show the reconstructed phase diagram, with each point corresponding to a sampled $(\langle\varphi\rangle, \eta)$ combination. The analytical spinodal line derived from the stability criterion of Eq. (6.9) is overlaid for reference. In the overwhelming majority of cases, 491 out of 500 (98.2%), the CRNN prediction matches the PF classification, correctly distinguishing between single-phase and two-phase configurations across

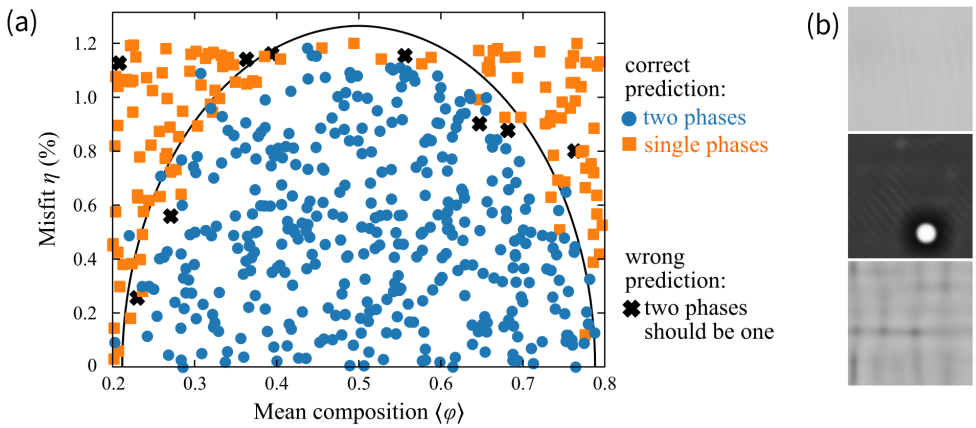

Figure 7.12: (a) Phase diagram reconstructed using the trained CRNN. Each point represents a $(\langle\varphi\rangle, \eta)$ pair sampled by PF simulations initialized from random Perlin noise and evolved up to $100, \tau$. The analytical spinodal line is shown as a solid black curve for reference, and points are classified based on the agreement between NN predictions and the corresponding PF ground truth. (b) Examples of failure cases at $200, \tau$, in which the model either incorrectly predicts phase separation or introduces spurious artifacts (indicated by black crosses in panel (a)).

the parametric space. Only 9 configurations resulted in misclassification, almost all occurring near the phase boundary between stability regimes. Fig. 7.12b illustrates three representative failure cases at $t = 200\,\tau$ where the CRNN incorrectly predicts phase separation in systems that should remain homogeneous according to ground-truth simulations. In these cases, the model introduces spurious compositional artifacts or weak phase separation features that do not appear in the true evolution.

This high degree of agreement confirms that the model has implicitly learned how elastic strain penalizes spinodal decomposition, shifting the stability boundary compared to the strain-free case, despite having no explicit knowledge of the analytical stability condition encoded in Eq. (6.9). Such implicit learning leads us to believe in the potential for applying this approach to more complex material systems where analytical expressions for phase stability are unavailable or where extensive parametric exploration would be computationally prohibitive using conventional PF methods alone.

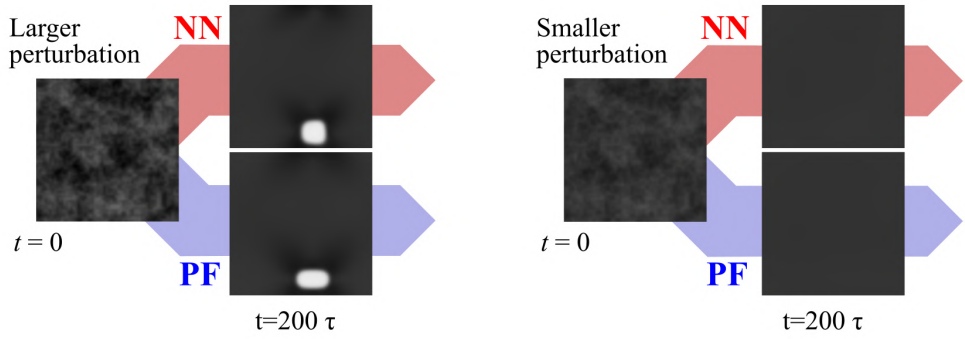

Figure 7.13: Comparison of NN and PF profiles at $200, \tau$ for two simulations starting from the same Perlin noise profile but with different amplitudes. The larger-amplitude perturbation (left) leads to phase separation, while the smaller one (right) remains homogeneous; both behaviors are correctly reproduced by the CRNN.

Sensitivity to Initial Perturbation Amplitude

An important caveat in interpreting the phase diagram is that average composition $\langle\varphi\rangle$ and misfit strain η alone do not uniquely determine the system's fate: the amplitude of initial perturbations also plays a significant role, particularly near the spinodal boundary. This explains why neither the CRNN nor the PF simulations produce a perfectly sharp phase boundary matching the analytical prediction. Both approaches exhibit some scatter around the theoretical curve due to variability in the amplitude of the perturbations in the initial conditions. To illustrate this sensitivity, in Fig. 7.13 we compare two evolutions initialized from the same Perlin noise profile but with different perturbation amplitudes from the mean value of the field. The configuration with larger fluctuations undergoes spinodal decomposition, developing a distinct phase-separated domain, while the one with smaller fluctuations relaxes to a uniform field. The CRNN accurately captures both outcomes, demonstrating that it has not just learned only the parametric dependence on $(\langle\varphi\rangle, \eta)$ but also the role of initial perturbation strength in determining nucleation dynamics.

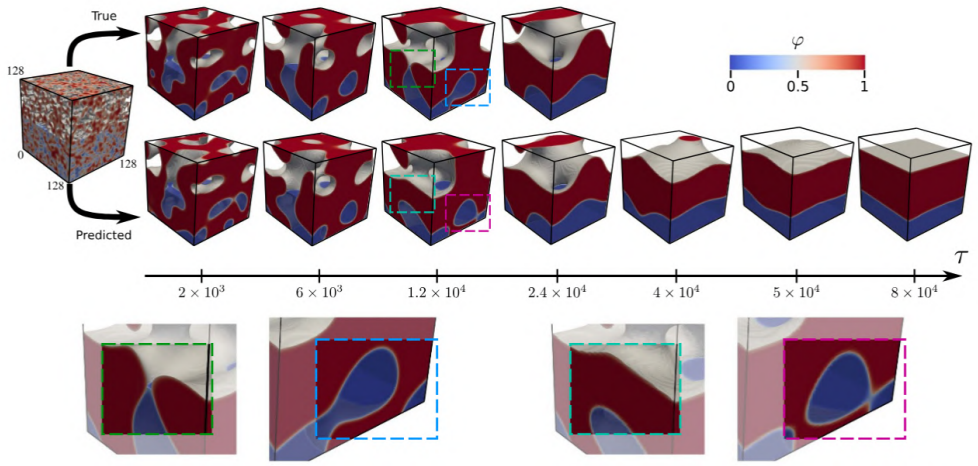

Figure 7.14: A domain initialized with Perlin random noise is evolved until it reaches a flat configuration. The evolution is compared with the corresponding finite difference integration for approximately one-third of the time sequence. In the upper set of snapshots, only regions with $\varphi \geq 0.5$ are displayed. Insets show close-ups of areas exhibiting the largest local deviations. Taken from our work, Ref. [20].

7.6 Extension to 3D domains

One of the main future objectives is to extend this line of research to three-dimensional domains. In the absence of elastic effects and anisotropy, such an extension is relatively straightforward. In fact, I already contributed to related work reported in Ref. [20], where a CRNN is trained to reproduce the three-dimensional spinodal decomposition dynamics governed by the Cahn–Hilliard equation. A specialized, physics-informed architecture is shown to yield excellent agreement between the predicted evolutions and the ground truth obtained from conventional numerical integration schemes. The approach accurately reproduces the evolution of microstructures not included in the training set, while achieving a substantial reduction in computational cost. As illustrated in Fig. 7.14, the model exhibits remarkable long-time extrapolation capabilities, reaching the theoretically expected equilibrium state characterized by a layered, phase-separated morphology, despite being trained only on short early stage dynamics. Quantitative agreement with the free energy decay rate is maintained up to the late coarsening regime, demonstrating that this class of machine learning methods can provide a powerful and thermodynamically consistent tool for long timescale and high-throughput ma-

terials simulations with high accuracy.

In expanding such a framework for systems with anisotropic elastic fields, no fundamental technical obstacles are expected: one can just pass the misfit parameter as a 3-dimensional uniform field, similarly to what has been done in this work in 2-dimensions. However, the construction of a sufficiently large and representative training database becomes more demanding.

Conclusions & Perspectives

About the MLIP model

Using the DeePMD framework [25], we have developed an accurate and computationally efficient machine learning interatomic potential, trained on a dataset of DFT calculations we performed using VASP, specifically designed to investigate pressure-driven crystal phase transitions in germanium. The resulting deep potential achieves several orders of magnitude speedup over density functional theory while retaining linear scaling with system size. Its reliability is demonstrated through extensive validation against DFT calculations, including regression analyses of energies, forces, and stresses on independent test sets, as well as accurate reproduction of formation energies and energy–volume relationships for a wide range of stable and metastable Ge allotropes.

A central outcome of this study is the demonstration that iterative, active refinement of the training dataset, by incorporating transition-state configurations sampled from solid-state-NEB and solid-state-dimer calculations performed with the potential itself, together with the training objective of fitting virials, together with energies, and forces, provides a simple yet highly effective strategy for achieving quantitatively accurate prediction of activation barriers under varying stress conditions. This capability enabled us to robustly explore several complex pressure-dependent transformation pathways, including the simulation of the dynamics of the pressure-induced nucleation of a crystal phase transition mechanism that would otherwise be prohibitively expensive to simulate with first-principles methods. A summary of such results is given at the very end of *PART II - Atomistic Modeling*.

To facilitate reproducibility and further research, the developed MLIP, together with the full reference dataset, has been made publicly available at Ref. [191].

This work provides a solid reference point for exploring several promising future research directions. First, the developed potential can be directly employed to investigate additional pressure-induced phase transi-

tions in germanium. In particular, the complete sequence of transformations connecting the cubic diamond and hexagonal diamond phases remains only partially understood. A natural extension of the present study would be to apply this potential to systematic investigations of these pathways, following approaches analogous to those previously adopted for silicon in Refs. [189, 6].

A second avenue of research involves extending the scope of the model to defect-mediated phenomena in germanium. The potential may be used to study dislocations, vacancies, and their mutual interactions, supported by targeted DFT calculations of representative defect configurations. Incorporating such structures into an expanded training dataset would enable further refinement of the potential and improve its transferability.

Alternatively, the existing high-quality dataset generated in this work could serve as a benchmark or training resource for next-generation machine learning potentials. For example, retraining the model using more recent architectures such as MACE [164] may yield further gains in accuracy or data efficiency. In a similar spirit, the dataset could be used to fine-tune modern foundation models, such as MACE-MP-0 [175, 176], and systematically assess their performance for pressure-driven phase transitions in germanium. While general-purpose models offer attractive flexibility, this comparison would help quantify the extent to which system-specific potentials continue to provide superior accuracy when sufficient domain-specific data are available.

About the CRNN model

We successfully developed and validated a parametrically-conditioned convolutional recurrent neural network surrogate for accelerating continuum scale simulations of coherent spinodal decomposition in alloys.

The surrogate model was trained on a dataset composed of time-evolution sequences of two-dimensional domains, generated by integrating the Cahn-Hilliard equation coupled with elasticity. After training, the model is provided with a given initial composition and an associated misfit parameter, and tasked with predicting a whole sequence iteratively. The model achieves near–perfect, one-to-one agreement with the phase-field ground truth across a broad range of misfit values and average compositions, within the temporal interpolation regime, corresponding to evolutions up to the training horizon of 100 τ (we recall that τ represents the time interval of each CRNN prediction and corresponds to $\tau = 100\ \delta t$, where δt is the phase-field simulation time step). When extrapolated to longer times, up to five times the training sequences,

pixel-level accuracy gradually degrades as small local errors accumulate and are amplified by bifurcation events such as domain coalescence and pinching. Despite this, the model continues to reliably reproduce global microstructural observables, including interface length, free-energy evolution, and geometric descriptors, over extended time scales.

Spatial generalization tests further demonstrate the scalability of the fully convolutional architecture, without retraining or architectural modification. This scalability yields significant computational advantages, thanks to a more favorable scaling with the number of collocation points ($\mathcal{O}(n)$ compared to the $\mathcal{O}(n \log n)$ complexity of the PF solver) when both the reference numerical simulations and model inference are executed on the CPU. Both the reference numerical solver and the NN inference can be accelerated on GPUs: the former via the CuPy library and the latter using PyTorch with CUDA support. When comparing GPU executions, the NN achieves roughly a $\times 3.3$ speedup relative to the reference numerical simulations. All benchmarks were performed on the same workstation equipped with an Intel Xeon W-2133 processor (12 cores at 3.60 GHz), 31 GB of RAM, and an NVIDIA RTX A5000 GPU.

In addition, the model robustly handles initial conditions that differ qualitatively from the training distribution, such as spatially varying composition gradients, underscoring its ability to generalize beyond narrowly defined data patterns.

The reconstruction of the phase diagram provides a stringent validation of the model's physical fidelity. Achieving a 98% accuracy in classifying homogeneous versus phase-separated states across the $(\langle\varphi\rangle, \eta)$ parameter space, the network has implicitly learned the thermodynamic principles governing phase stability, including the strain-induced suppression of spinodal decomposition, despite having no explicit access to analytical stability criteria. This result highlights the potential of the approach for systems where analytical descriptions are unavailable or exhaustive parametric studies would be computationally prohibitive.

Although the present study focused on spinodal decomposition described by the elastically coupled Cahn–Hilliard equation, the proposed framework is broadly applicable to other continuum models. This system was chosen as a demanding test case due to its rich morphological behavior, providing a rigorous benchmark for the architecture. The underlying methodological elements, including parametric conditioning, physics-informed output layers enforcing conservation laws, and multi-term loss functions balancing local accuracy with thermodynamic consistency, can be adapted to other phase-field and continuum formulations.

The combination of low computational cost and demonstrated scalability opens the door to high-throughput investigations of systems that lie beyond the practical reach of conventional numerical solvers. In particular, the ability to train on modest domain sizes and deploy on much larger systems offers a decisive advantage for studying phenomena characterized by large intrinsic length scales.

The dataset used to train and validate the model is openly available in Materials Cloud Archive Ref. [234]. The code used to train the NN model is freely available on GitHub at Ref. [235].

Looking ahead, it would be worth trying to implement other models based on novel, promising architectures like U-nets [217], DeepONets [220], and Fourier Neural Operators [236]. Also, an exciting direction for further development lies in training the surrogate model on experimental data. Incorporating real microstructural observations from techniques such as microscopy or scattering could enable hybrid, data-informed models that bridge simulations and experiments.

List of Publications

Work directly connected to the Thesis

A Fantasia, F Rovaris, O Abou El Kheir, A Marzegalli, D Lanzoni, L Pessina, P Xiao, C Zhou, L Li, G Henkelman, E Scalise, F Montalenti. *Development of a machine learning interatomic potential for exploring pressure-dependent kinetics of phase transitions in germanium.* The Journal of Chemical Physics 161 (1), 2024
[https://doi.org/10.1063/5.0214588]

D Lanzoni, **A Fantasia**, R Bergamaschini, O Pierre-Louis, F Montalenti. *Extreme time extrapolation capabilities and thermodynamic consistency of physics-inspired neural networks for the 3D microstructure evolution of materials via Cahn–Hilliard flow.* Machine Learning: Science and Technology 5 (4), 045017, 2024
[https://doi.org/10.1088/2632-2153/ad8532]

A Fantasia, D Lanzoni, N Di Eugenio, A Monteleone, R Bergamaschini, F Montalenti. *A parametrically-conditioned Deep Learning surrogate for coherent spinodal decomposition.* Under review for publication, 2026.

Further work

I contributed to collaborative research focused on developing novel convolutional neural network-based strategies to accelerate growth dynamics in strained thin films. The challenge addressed in these works is that the elastic contribution to the chemical potential, typically evaluated through Green's function approximations or full finite element calculations, represents the dominant computational bottleneck in the time integration of surface evolution equations, limiting long-time and large-scale simulations. We show that we can train CNNs to accurately ap-

proximate the mapping between surface morphology and elastic energy density. This approach enables fast and stable simulations of growth and coarsening dynamics, achieving substantial computational speedups without sacrificing physical accuracy. My main contribution consisted of developing and implementing the code used to generate large ensembles of suitably randomized initial film profiles for NN training and validation.

D Lanzoni, F Rovaris, L Martín-Encinar, **A Fantasia**, R Bergamaschini, F Montalenti. *Accelerating simulations of strained-film growth by deep learning: Finite element method accuracy over long time scales.* APL Machine Learning 2 (3), 2024
[https://doi.org/10.1063/5.0221363]

L Martín-Encinar, D Lanzoni, **A Fantasia**, F Rovaris, R Bergamaschini, F Montalenti. *Quantitative analysis of the prediction performance of a convolutional neural network evaluating the surface elastic energy of a strained film.* Computational Materials Science 249, 113657, 2025
[https://doi.org/10.1016/j.commatsci.2024.113657]

Bibliography

[1] F. H. Stillinger and T. A. Weber, "Computer simulation of local order in condensed phases of silicon," *Physical Review B*, vol. 31, pp. 5262–5271, Apr. 1985.

[2] K. Ding and H. C. Andersen, "Molecular-dynamics simulation of amorphous germanium," *Physical Review B*, vol. 34, pp. 6987–6991, Nov. 1986.

[3] J. Tersoff, "Modeling solid-state chemistry: Interatomic potentials for multicomponent systems," *Physical Review B*, vol. 39, pp. 5566–5568, Mar. 1989.

[4] M. I. Baskes, J. S. Nelson, and A. F. Wright, "Semiempirical modified embedded-atom potentials for silicon and germanium," *Physical Review B*, vol. 40, pp. 6085–6100, Sept. 1989.

[5] E. H. Kim, Y.-H. Shin, and B.-J. Lee, "A modified embedded-atom method interatomic potential for Germanium," *Calphad*, vol. 32, pp. 34–42, Mar. 2008.

[6] G. Ge, F. Rovaris, D. Lanzoni, L. Barbisan, X. Tang, L. Miglio, A. Marzegalli, E. Scalise, and F. Montalenti, "Silicon phase transitions in nanoindentation: Advanced molecular dynamics simulations with machine learning phase recognition," *Acta Materialia*, vol. 263, p. 119465, Jan. 2024.

[7] J. Behler and M. Parrinello, "Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces," *Physical Review Letters*, vol. 98, p. 146401, Apr. 2007.

[8] J. Behler, "Neural network potential-energy surfaces in chemistry: a tool for large-scale simulations," *Physical Chemistry Chemical Physics*, vol. 13, no. 40, p. 17930, 2011.

[9] A. V. Shapeev, "Moment Tensor Potentials: A Class of Systematically Improvable Interatomic Potentials," *Multiscale Modeling & Simulation*, vol. 14, pp. 1153–1173, Jan. 2016.

[10] A. P. Bartók, J. Kermode, N. Bernstein, and G. Csányi, "Machine Learning a General-Purpose Interatomic Potential for Silicon," *Physical Review X*, vol. 8, p. 041048, Dec. 2018.

[11] Y. Zuo, C. Chen, X. Li, Z. Deng, Y. Chen, J. Behler, G. Csányi, A. V. Shapeev, A. P. Thompson, M. A. Wood, and S. P. Ong, "Performance and Cost Assessment of Machine Learning Interatomic Potentials," *The Journal of Physical Chemistry A*, vol. 124, pp. 731–745, Jan. 2020.

[12] Y. Lysogorskiy, C. V. D. Oord, A. Bochkarev, S. Menon, M. Rinaldi, T. Hammerschmidt, M. Mrovec, A. Thompson, G. Csányi, C. Ortner, and R. Drautz, "Performant implementation of the atomic cluster expansion (PACE) and application to copper and silicon," *npj Computational Materials*, vol. 7, p. 97, June 2021.

[13] O. Abou El Kheir and M. Bernasconi, "Million-Atom Simulation of the Set Process in Phase Change Memories at the Real Device Scale," *Advanced Electronic Materials*, vol. 11, p. e2500110, Aug. 2025.

[14] N. Provatas and K. Elder, *Phase-Field Methods in Materials Science and Engineering*. Wiley, 1 ed., Oct. 2010.

[15] D. Lanzoni, F. Montalenti, and R. Bergamaschini, "Deep learning for simulating the evolution of condensed matter systems at the continuum scale: Methods and applications," *Journal of Physics: Condensed Matter*, vol. 37, p. 403003, 2025.

[16] K. Alhada-Lahbabi, D. Deleruyelle, and B. Gautier, "Machine learning surrogate for 3d phase-field modeling of ferroelectric tip-induced electrical switching," *npj Computational Materials*, vol. 10, p. 197, 2024.

[17] K. Yang, Y. Cao, Y. Zhang, S. Fan, M. Tang, D. Aberg, B. Sadigh, and F. Zhou, "Self-supervised learning and prediction of microstructure evolution with convolutional recurrent neural networks," *Patterns*, vol. 2, p. 100243, May 2021.

[18] D. Montes De Oca Zapiain, J. A. Stewart, and R. Dingreville, "Accelerating phase-field-based microstructure evolution predictions via surrogate models trained by machine learning methods," *npj Comput Mater*, vol. 7, p. 3, Jan. 2021.

[19] A. A. Kazemzadeh Farizhandi and M. Mamivand, "Spatiotemporal prediction of microstructure evolution with predictive recurrent neural network," *Computational Materials Science*, vol. 223, p. 112110, 2023.

[20] D. Lanzoni, A. Fantasia, R. Bergamaschini, O. Pierre-Louis, and F. Montalenti, "Extreme time extrapolation capabilities and thermodynamic consistency of physics-inspired neural networks for the 3D microstructure evolution of materials via Cahn–Hilliard flow," *Mach. Learn.: Sci. Technol.*, vol. 5, p. 045017, Dec. 2024.

[21] V. Oommen, K. Shukla, S. Desai, R. Dingreville, and G. E. Karniadakis, "Rethinking materials simulations: Blending direct numerical simulations with neural operators," 2023. Version Number: 1.

[22] E. M. T. Fadaly, A. Dijkstra, J. R. Suckert, D. Ziss, M. A. J. Van Tilburg, C. Mao, Y. Ren, V. T. Van Lange, K. Korzun, S. Kölling, M. A. Verheijen, D. Busse, C. Rödl, J. Furthmüller, F. Bechstedt, J. Stangl, J. J. Finley, S. Botti, J. E. M. Haverkort, and E. P. A. M. Bakkers, "Direct-bandgap emission from hexagonal Ge and SiGe alloys," *Nature*, vol. 580, pp. 205–209, Apr. 2020.

[23] L. Zhang, J. Han, H. Wang, R. Car, and W. E, "Deep Potential Molecular Dynamics: A Scalable Model with the Accuracy of Quantum Mechanics," *Physical Review Letters*, vol. 120, p. 143001, Apr. 2018.

[24] L. Zhang, J. Han, H. Wang, W. Saidi, R. Car, and W. E, "End-to-end Symmetry Preserving Inter-atomic Potential Energy Model for Finite and Extended Systems," in *Advances in Neural Information Processing Systems* (S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds.), vol. 31, Curran Associates, Inc., 2018.

[25] H. Wang, L. Zhang, J. Han, and W. E, "DeePMD-kit: A deep learning package for many-body potential energy representation

and molecular dynamics," *Computer Physics Communications*, vol. 228, pp. 178–184, July 2018.

[26] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," 2019. Version Number: 1.

[27] A. I. J. Forrester, A. Sóbester, and A. J. Keane, *Engineering Design via Surrogate Modelling: A Practical Guide*. Wiley, 1 ed., July 2008.

[28] F. Campolongo, J. Cariboni, and A. Saltelli, "An effective screening design for sensitivity analysis of large models," *Environmental Modelling & Software*, vol. 22, pp. 1509–1518, Oct. 2007.

[29] H. Kapadia, L. Feng, and P. Benner, "Active-learning-driven surrogate modeling for efficient simulation of parametric nonlinear systems," *Computer Methods in Applied Mechanics and Engineering*, vol. 419, p. 116657, Feb. 2024.

[30] M. Balesdent, L. Brevaul, S. Lacaze, S. Missoum, and J. Morio, "Methods for high-dimensional and computationally intensive models," in *Estimation of Rare Event Probabilities in Complex Aerospace and Other Systems*, pp. 109–136, Elsevier, 2016.

[31] J. Helton and F. Davis, "Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems," *Reliability Engineering & System Safety*, vol. 81, pp. 23–69, July 2003.

[32] C. D. Lin and B. Tang, "Latin Hypercubes and Space-filling Designs," 2022. Publisher: arXiv Version Number: 1.

[33] V. R. Joseph, "Space-filling designs for computer experiments: A review," *Quality Engineering*, vol. 28, pp. 28–35, Jan. 2016.

[34] X. Sun, B. Croke, S. Roberts, and A. Jakeman, "Comparing methods of randomizing Sobol sequences for improving uncertainty of metrics in variance-based global sensitivity estimation," *Reliability Engineering & System Safety*, vol. 210, p. 107499, June 2021.

[35] H. Niederreiter, *Random number generation and quasi-Monte Carlo methods*. SIAM, 1992.

[36] D. Gorissen, I. Couckuyt, P. Demeester, T. Dhaene, and K. Crombecq, "A surrogate modeling and adaptive sampling toolbox for computer based design," *Journal of Machine Learning Research*, vol. 11, pp. 2051–2055, 07 2010.

[37] P. Westermann and R. Evins, "Adaptive sampling for building simulation surrogate model derivation using the lola-voronoi algorithm," pp. 1559–1563, 01 2019.

[38] H. Li, S. Sen, and L. Khazanovich, "A scalable adaptive sampling approach for surrogate modeling of rigid pavements using machine learning," *Results in Engineering*, vol. 23, p. 102483, Sept. 2024.

[39] X. Wang, K. Tang, J. Zhai, X. Wan, and C. Yang, "Deep adaptive sampling for surrogate modeling without labeled data," 2024. Publisher: arXiv Version Number: 1.

[40] Ben-Israel Greville 2003, p. 7 Campbell Meyer 1991, p. 10 Nakamura 1991, p. 42 Rao Mitra 1971, p. 50–51.

[41] A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi, "Gaussian Approximation Potentials: The Accuracy of Quantum Mechanics, without the Electrons," *Phys. Rev. Lett.*, vol. 104, p. 136403, Apr. 2010.

[42] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain.," *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958.

[43] K. Hornik, M. Stinchcombe, and H. White, "Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks," *Neural Networks*, vol. 3, pp. 551–560, Jan. 1990.

[44] J. Heaton, "Ian goodfellow, yoshua bengio, and aaron courville: Deep learning: The mit press, 2016, 800 pp, isbn: 0262035618," *Genetic Programming and Evolvable Machines*, vol. 19, 10 2017.

[45] G. Hinton, V. Nair, and Y. Rachmad, "Rectified linear units improve restricted boltzmann machines vinod nair," vol. 27, pp. 807–814, 06 2010.

[46] A. L. "Maas, A. Y. "Hannun, and A. Y. "Ng, "Rectifier nonlinearities improve neural network acoustic models," 2013.

[47] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *Under Review of ICLR2016 (1997)*, 11 2015.

[48] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," 2023.

[49] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," vol. 15, 01 2010.

[50] P. J. Huber, "Robust Estimation of a Location Parameter," *The Annals of Mathematical Statistics*, vol. 35, pp. 73–101, Mar. 1964.

[51] L. Bottou and O. Bousquet, "The tradeoffs of large scale learning.," vol. 20, 01 2007.

[52] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. 61, pp. 2121–2159, 2011.

[53] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," 2014.

[54] M. Zaheer, S. J. Reddi, D. Sachan, S. Kale, and S. Kumar, "Adaptive methods for nonconvex optimization," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, (Red Hook, NY, USA), p. 9815–9825, Curran Associates Inc., 2018.

[55] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," 2012.

[56] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.

[57] D. Warde-Farley, I. J. Goodfellow, A. Courville, and Y. Bengio, "An empirical analysis of dropout in piecewise linear networks," 2014.

[58] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015.

[59] J. Behler, "Atom-centered symmetry functions for constructing high-dimensional neural network potentials," *The Journal of Chemical Physics*, vol. 134, p. 074106, Feb. 2011.

[60] R. Jacobs, D. Morgan, S. Attarian, J. Meng, C. Shen, Z. Wu, C. Y. Xie, J. H. Yang, N. Artrith, B. Blaiszik, G. Ceder, K. Choudhary, G. Csanyi, E. D. Cubuk, B. Deng, R. Drautz, X. Fu, J. Godwin, V. Honavar, O. Isayev, A. Johansson, B. Kozinsky, S. Martiniani, S. P. Ong, I. Poltavsky, K. Schmidt, S. Takamoto, A. P. Thompson, J. Westermayr, and B. M. Wood, "A practical guide to machine learning interatomic potentials – Status and future," *Current Opinion in Solid State and Materials Science*, vol. 35, p. 101214, Mar. 2025.

[61] G. Voronin, C. Pantea, T. Zerda, J. Zhang, L. Wang, and Y. Zhao, "In situ X-ray diffraction study of germanium at pressures up to 11 GPa and temperatures up to 950K," *Journal of Physics and Chemistry of Solids*, vol. 64, pp. 2113–2119, Nov. 2003.

[62] A. Mujica, A. Rubio, A. Muñoz, and R. J. Needs, "High-pressure phases of group-IV, III–V, and II–VI compounds," *Reviews of Modern Physics*, vol. 75, pp. 863–912, July 2003.

[63] F. Coppari, J. C. Chervin, A. Congeduti, M. Lazzeri, A. Polian, E. Principi, and A. Di Cicco, "Pressure-induced phase transitions in amorphous and metastable crystalline germanium by Raman scattering, x-ray spectroscopy, and *ab initio* calculations," *Physical Review B*, vol. 80, p. 115213, Sept. 2009.

[64] K. Takemura, U. Schwarz, K. Syassen, M. Hanfland, N. E. Christensen, D. L. Novikov, and I. Loa, "High-pressure Cmca and hcp phases of germanium," *Physical Review B*, vol. 62, pp. R10603–R10606, Oct. 2000.

[65] X.-J. Chen, C. Zhang, Y. Meng, R.-Q. Zhang, H.-Q. Lin, V. V. Struzhkin, and H.-k. Mao, "β-tin -¿ Imma -¿ sh Phase Transitions of Germanium," *Physical Review Letters*, vol. 106, p. 135502, Mar. 2011.

[66] J.-i. Jang, M. J. Lance, S. Wen, and G. M. Pharr, "Evidence for nanoindentation-induced phase transformations in germanium," *Applied Physics Letters*, vol. 86, p. 131907, Mar. 2005.

[67] Z. Zhao, H. Zhang, D. Y. Kim, W. Hu, E. S. Bullock, and T. A. Strobel, "Properties of the exotic metastable ST12 germanium allotrope," *Nature Communications*, vol. 8, p. 13909, Jan. 2017.

[68] L. Q. Huston, B. C. Johnson, B. Haberl, S. Wong, J. S. Williams, and J. E. Bradby, "Thermal stability of simple tetragonal and hexagonal diamond germanium," *Journal of Applied Physics*, vol. 122, p. 175108, Nov. 2017.

[69] B. C. Johnson, B. Haberl, S. Deshmukh, B. D. Malone, M. L. Cohen, J. C. McCallum, J. S. Williams, and J. E. Bradby, "Evidence for the R 8 Phase of Germanium," *Physical Review Letters*, vol. 110, p. 085502, Feb. 2013.

[70] K. Kosai, H. Huang, and J. Yan, "Comparative Study of Phase Transformation in Single-Crystal Germanium during Single and Cyclic Nanoindentation," *Crystals*, vol. 7, p. 333, Nov. 2017.

[71] B. D. Malone and M. L. Cohen, "Electronic structure, equation of state, and lattice dynamics of low-pressure Ge polymorphs," *Physical Review B*, vol. 86, p. 054101, Aug. 2012.

[72] Q. Yuan, S. Li, L. Zhou, and D. He, "Phase-pure ST12 Ge bulks through secondary pressure induced phase transition," *Solid State Communications*, vol. 348-349, p. 114742, June 2022.

[73] B. Haberl, M. Guthrie, B. D. Malone, J. S. Smith, S. V. Sinogeikin, M. L. Cohen, J. S. Williams, G. Shen, and J. E. Bradby, "Controlled formation of metastable germanium polymorphs," *Physical Review B*, vol. 89, p. 144111, Apr. 2014.

[74] S. Tarkhorani and M. Sasani Ghamsari, "Novel route for preparation of ST12-Ge nanoparticles at ambient pressure," *Materials Science and Engineering: B*, vol. 261, p. 114665, Nov. 2020.

[75] A. Garcia-Gil, S. Biswas, A. Roy, D. Saladukh, S. Raha, T. Blon, M. Conroy, V. Nicolosi, A. Singha, L.-M. Lacroix, and J. D. Holmes, "Growth and analysis of the tetragonal (ST12) germanium nanowires," *Nanoscale*, vol. 14, no. 5, pp. 2030–2040, 2022.

[76] J. S. Williams, B. Haber, S. Deshmukh, B. C. Johnson, B. D. Malone, M. L. Cohen, and J. E. Bradby, "Hexagonal germanium formed via a pressure-induced phase transformation of amorphous germanium under controlled nanoindentation," *physica status solidi (RRL) – Rapid Research Letters*, vol. 7, pp. 355–359, May 2013.

[77] G. Dushaq, A. Nayfeh, and M. Rasras, "Hexagonal germanium formation at room temperature using controlled penetration depth nano-indentation," *Scientific Reports*, vol. 9, p. 1593, Feb. 2019.

[78] A. G. Lyapin, V. V. Brazhkin, S. V. Popova, and A. V. Sapelkin, "Nonequilibrium Phase Transformations in Diamond and Zincblende Semiconductors under High Pressure," *physica status solidi (b)*, vol. 198, pp. 481–490, Nov. 1996.

[79] B. Haberl, J. E. Bradby, M. V. Swain, J. S. Williams, and P. Munroe, "Phase transformations induced in relaxed amorphous silicon by indentation at room temperature," *Applied Physics Letters*, vol. 85, pp. 5559–5561, Dec. 2004.

[80] A. Di Cicco, A. Congeduti, F. Coppari, J. C. Chervin, F. Baudelet, and A. Polian, "Interplay between morphology and metallization in amorphous-amorphous transitions," *Physical Review B*, vol. 78, p. 033309, July 2008.

[81] M. Born and R. Oppenheimer, "Zur Quantentheorie der Molekeln," *Annalen der Physik*, vol. 389, pp. 457–484, Jan. 1927.

[82] P. Hohenberg and W. Kohn, "Inhomogeneous Electron Gas," *Physical Review*, vol. 136, pp. B864–B871, Nov. 1964.

[83] W. Kohn and L. J. Sham, "Self-Consistent Equations Including Exchange and Correlation Effects," *Physical Review*, vol. 140, pp. A1133–A1138, Nov. 1965.

[84] D. C. Langreth and M. J. Mehl, "Beyond the local-density approximation in calculations of ground-state electronic properties," *Physical Review B*, vol. 28, pp. 1809–1834, Aug. 1983.

[85] A. D. Becke, "Density-functional exchange-energy approximation with correct asymptotic behavior," *Physical Review A*, vol. 38, pp. 3098–3100, Sept. 1988.

[86] C. Adamo, M. Ernzerhof, and G. E. Scuseria, "The meta-GGA functional: Thermochemistry with a kinetic energy density dependent exchange-correlation functional," *The Journal of Chemical Physics*, vol. 112, pp. 2643–2649, Feb. 2000.

[87] L. Simón and J. M. Goodman, "How reliable are DFT transition structures? Comparison of GGA, hybrid-meta-GGA and meta-GGA functionals," *Org. Biomol. Chem.*, vol. 9, no. 3, pp. 689–700, 2011.

[88] Y. Zhang, J. Sun, J. P. Perdew, and X. Wu, "Comparative first-principles studies of prototypical ferroelectric materials by LDA, GGA, and SCAN meta-GGA," *Physical Review B*, vol. 96, p. 035143, July 2017.

[89] D. Mejía-Rodríguez and S. B. Trickey, "Meta-GGA performance in solids at almost GGA cost," *Physical Review B*, vol. 102, p. 121109, Sept. 2020.

[90] D. R. Hamann, M. Schlüter, and C. Chiang, "Norm-conserving pseudopotentials," *Phys. Rev. Lett.*, vol. 43, pp. 1494–1497, Nov 1979.

[91] H. J. Monkhorst and J. D. Pack, "Special points for brillouin-zone integrations," *Phys. Rev. B*, vol. 13, pp. 5188–5192, Jun 1976.

[92] D. S. Sholl and J. A. Steckel, *Density Functional Theory: A Practical Introduction*. Hoboken, NJ: John Wiley & Sons, 2009.

[93] E. Engel and R. M. Dreizler, *Density Functional Theory: An Advanced Course*. Theoretical and Mathematical Physics, Berlin, Heidelberg: Springer, 2011.

[94] S. Hussain and A. Haji-Akbari, "Studying rare events using forward-flux sampling: Recent breakthroughs and future outlook," *The Journal of Chemical Physics*, vol. 152, p. 060901, 02 2020.

[95] S. W. Hall, G. Díaz Leines, S. Sarupria, and J. Rogal, "Practical guide to replica exchange transition interface sampling and forward flux sampling," *The Journal of Chemical Physics*, vol. 156, May 2022.

[96] K. Ahmad, A. Rizzi, R. Capelli, D. Mandelli, W. Lyu, and P. Carloni, "Enhanced-Sampling Simulations for the Estimation of Ligand Binding Kinetics: Current Status and Perspective," *Frontiers in Molecular Biosciences*, vol. 9, p. 899805, June 2022.

[97] G. Torrie and J. Valleau, "Nonphysical sampling distributions in Monte Carlo free-energy estimation: Umbrella sampling," *Journal of Computational Physics*, vol. 23, pp. 187–199, Feb. 1977.

[98] F. Wang and D. P. Landau, "Efficient, Multiple-Range Random Walk Algorithm to Calculate the Density of States," *Physical Review Letters*, vol. 86, pp. 2050–2053, Mar. 2001.

[99] E. Darve and A. Pohorille, "Calculating free energies using average force," *The Journal of Chemical Physics*, vol. 115, pp. 9169–9183, Nov. 2001.

[100] A. Laio and M. Parrinello, "Escaping free-energy minima," *Proceedings of the National Academy of Sciences*, vol. 99, pp. 12562–12566, Oct. 2002.

[101] L. Rosso, P. Mináry, Z. Zhu, and M. E. Tuckerman, "On the use of the adiabatic molecular dynamics technique in the calculation of free energy profiles," *The Journal of Chemical Physics*, vol. 116, pp. 4389–4402, Mar. 2002.

[102] L. Maragliano and E. Vanden-Eijnden, "A temperature accelerated method for sampling free energy and determining reaction pathways in rare events simulations," *Chemical Physics Letters*, vol. 426, pp. 168–175, July 2006.

[103] G. M. Torrie and J. P. Valleau, "Monte Carlo free energy estimates using non-Boltzmann sampling: Application to the subcritical Lennard-Jones fluid," *Chemical Physics Letters*, vol. 28, pp. 578–581, Oct. 1974.

[104] A. Barducci, G. Bussi, and M. Parrinello, "Well-Tempered Metadynamics: A Smoothly Converging and Tunable Free-Energy Method," *Physical Review Letters*, vol. 100, p. 020603, Jan. 2008.

[105] M. Invernizzi and M. Parrinello, "Rethinking Metadynamics: From Bias Potentials to Probability Distributions," *The Journal of Physical Chemistry Letters*, vol. 11, pp. 2731–2736, Apr. 2020.

[106] D. Ray, N. Ansari, V. Rizzi, M. Invernizzi, and M. Parrinello, "Rare Event Kinetics from Adaptive Bias Enhanced Sampling," *Journal of Chemical Theory and Computation*, vol. 18, pp. 6500–6509, Nov. 2022.

[107] T. S. Van Erp, "Efficient path sampling on multiple reaction channels," *Computer Physics Communications*, vol. 179, pp. 34–40, July 2008.

[108] A. K. Faradjian and R. Elber, "Computing time scales from reaction coordinates by milestoning," *The Journal of Chemical Physics*, vol. 120, pp. 10880–10889, June 2004.

[109] A. M. A. West, R. Elber, and D. Shalloway, "Extending molecular dynamics time scales with milestoning: Example of complex kinetics in a solvated peptide," *The Journal of Chemical Physics*, vol. 126, p. 145104, Apr. 2007.

[110] R. J. Allen, D. Frenkel, and P. R. Ten Wolde, "Simulating rare events in equilibrium or nonequilibrium stochastic systems," *The Journal of Chemical Physics*, vol. 124, p. 024102, Jan. 2006.

[111] D. Ray and I. Andricioaei, "Weighted ensemble milestoning (WEM): A combined approach for rare event simulations," *The Journal of Chemical Physics*, vol. 152, p. 234114, June 2020.

[112] D. Ray, S. E. Stone, and I. Andricioaei, "Markovian Weighted Ensemble Milestoning (M-WEM): Long-Time Kinetics from Short Trajectories," *Journal of Chemical Theory and Computation*, vol. 18, pp. 79–95, Jan. 2022.

[113] P. G. Bolhuis, D. Chandler, C. Dellago, and P. L. Geissler, "Transition path sampling: throwing ropes over rough mountain passes, in the dark," *Annual Review of Physical Chemistry*, vol. 53, pp. 291–318, Oct. 2002.

[114] H. Jung, R. Covino, A. Arjun, C. Leitold, C. Dellago, P. G. Bolhuis, and G. Hummer, "Machine-guided path sampling to discover mechanisms of molecular self-organization," *Nature Computational Science*, vol. 3, pp. 334–345, Apr. 2023.

[115] S. Asghar, Q.-X. Pei, G. Volpe, and R. Ni, "Efficient rare event sampling with unsupervised normalizing flows," *Nature Machine Intelligence*, vol. 6, pp. 1370–1381, Nov. 2024.

[116] F. Cichos, K. Gustavsson, B. Mehlig, and G. Volpe, "Machine learning for active matter," *Nature Machine Intelligence*, vol. 2, pp. 94–103, Feb. 2020.

[117] M. Gabrié, G. M. Rotskoff, and E. Vanden-Eijnden, "Adaptive Monte Carlo augmented with normalizing flows," *Proceedings of the National Academy of Sciences*, vol. 119, p. e2109420119, Mar. 2022.

[118] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep Unsupervised Learning using Nonequilibrium Thermodynamics," 2015. Version Number: 8.

[119] J. Ho, A. Jain, and P. Abbeel, "Denoising Diffusion Probabilistic Models," 2020. Version Number: 2.

[120] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-Based Generative Modeling through Stochastic Differential Equations," 2020. Version Number: 2.

[121] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical Text-Conditional Image Generation with CLIP Latents," 2022. Version Number: 1.

[122] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models," 2021. Version Number: 2.

[123] M. Petersen, G. Roig, and R. Covino, "Dynamicsdiffusion: Generating and rare event sampling of molecular dynamic trajectories using diffusion models," in *NeurIPS 2023 AI for Science Workshop*, 2023.

[124] R. E. Gillilan and K. R. Wilson, "Shadowing, rare events, and rubber bands. a variational verlet algorithm for molecular dynamics," *The Journal of Chemical Physics*, vol. 97, pp. 1757–1772, 08 1992.

[125] W. Ee and E. Vanden-Eijnden, "Towards a theory of transition paths," *Journal of Statistical Physics - J STATIST PHYS*, vol. 123, pp. 503–523, 05 2006.

[126] W. E and E. Vanden-Eijnden, "Transition-Path Theory and Path-Finding Algorithms for the Study of Rare Events," *Annual Review of Physical Chemistry*, vol. 61, pp. 391–420, Mar. 2010.

[127] G. Henkelman, B. P. Uberuaga, and H. Jónsson, "A climbing image nudged elastic band method for finding saddle points and minimum energy paths," *The Journal of Chemical Physics*, vol. 113, pp. 9901–9904, Dec. 2000.

[128] G. Henkelman and H. Jónsson, "Improved tangent estimate in the nudged elastic band method for finding minimum energy paths and saddle points," *The Journal of Chemical Physics*, vol. 113, pp. 9978–9985, Dec. 2000.

[129] G. Henkelman and H. Jónsson, "A dimer method for finding saddle points on high dimensional potential surfaces using only first derivatives," *The Journal of Chemical Physics*, vol. 111, pp. 7010–7022, Oct. 1999.

[130] I. Baghishov, "Computational chemistry and materials science dft post." `https://www.linkedin.com/posts/ilgarbaghishov_computationalchemistry-materialsscience-dft-activity-7407127709929213952-h6-B`, 2025.

[131] R. A. Olsen, G. J. Kroes, G. Henkelman, A. Arnaldsson, and H. Jónsson, "Comparison of methods for finding saddle points without knowledge of the final states," *The Journal of Chemical Physics*, vol. 121, pp. 9776–9792, 11 2004.

[132] D. Sheppard, R. Terrell, and G. Henkelman, "Optimization methods for finding minimum energy paths," *The Journal of Chemical Physics*, vol. 128, p. 134106, 04 2008.

[133] D. Sheppard, P. Xiao, W. Chemelewski, D. D. Johnson, and G. Henkelman, "A generalized solid-state nudged elastic band method," *The Journal of Chemical Physics*, vol. 136, p. 074103, Feb. 2012.

[134] P. Xiao, D. Sheppard, J. Rogal, and G. Henkelman, "Solid-state dimer method for calculating solid-solid phase transitions," *The Journal of Chemical Physics*, vol. 140, p. 174104, 05 2014.

[135] J. P. Perdew, K. Burke, and M. Ernzerhof, "Generalized gradient approximation made simple," *Phys. Rev. Lett.*, vol. 77, pp. 3865–3868, Oct 1996.

[136] A. P. Thompson, H. M. Aktulga, R. Berger, D. S. Bolintineanu, W. M. Brown, P. S. Crozier, P. J. in 't Veld, A. Kohlmeyer, S. G. Moore, T. D. Nguyen, R. Shan, M. J. Stevens, J. Tranchida, C. Trott, and S. J. Plimpton, "Lammps - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales," *Computer Physics Communications*, vol. 271, p. 108171, 2022.

[137] E. Bitzek, P. Koskinen, F. Gähler, M. Moseler, and P. Gumbsch, "Structural relaxation made simple," *Phys. Rev. Lett.*, vol. 97, p. 170201, Oct 2006.

[138] A. Hjorth Larsen, J. Jørgen Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dułak, J. Friis, M. N. Groves, B. Hammer, C. Hargus, E. D. Hermes, P. C. Jennings, P. Bjerre Jensen, J. Kermode, J. R. Kitchin, E. Leonhard Kolsbjerg, J. Kubal, K. Kaasbjerg, S. Lysgaard, J. Bergmann Maronsson, T. Maxson, T. Olsen, L. Pastewka, A. Peterson, C. Rostgaard, J. Schiøtz, O. Schütt, M. Strange, K. S. Thygesen, T. Vegge, L. Vilhelmsen, M. Walter, Z. Zeng, and K. W. Jacobsen, "The atomic simulation environment—a python library for working with atoms," *Journal of Physics: Condensed Matter*, vol. 29, p. 273002, jun 2017.

[139] http://theory.cm.utexas.edu/henkelman/code/. To obtain the TSASE code.

[140] A. P. Bartók, R. Kondor, and G. Csányi, "On representing chemical environments," *Phys. Rev. B*, vol. 87, p. 184115, May 2013.

[141] R. Drautz, "Atomic cluster expansion for accurate and transferable interatomic potentials," *Phys. Rev. B*, vol. 99, p. 014104, Jan 2019.

[142] A. Thompson, L. Swiler, C. Trott, S. Foiles, and G. Tucker, "Spectral neighbor analysis method for automated generation of quantum-accurate interatomic potentials," *Journal of Computational Physics*, vol. 285, pp. 316–330, 2015.

[143] A. V. Shapeev, "Moment tensor potentials: A class of systematically improvable interatomic potentials," *Multiscale Modeling & Simulation*, vol. 14, no. 3, pp. 1153–1173, 2016.

[144] K. Nguyen-Cong, J. T. Willman, S. G. Moore, A. B. Belonoshko, R. Gayatri, E. Weinberg, M. A. Wood, A. P. Thompson, and I. I.

Oleynik, "Billion atom molecular dynamics simulations of carbon at extreme conditions and experimental time and length scales," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '21, (New York, NY, USA), Association for Computing Machinery, 2021.

[145] T. B. Blank, S. D. Brown, A. W. Calhoun, and D. J. Doren, "Neural network models of potential energy surfaces," *The Journal of Chemical Physics*, vol. 103, pp. 4129–4137, Sept. 1995.

[146] X. Wang, Y. Wang, L. Zhang, F. Dai, and H. Wang, "A tungsten deep neural-network potential for simulating mechanical property degradation under fusion service environment," *Nuclear Fusion*, vol. 62, p. 126013, Dec. 2022.

[147] H. Weyl, *The Classical Groups: Their Invariants and Representations*. Princeton University Press, June 2016.

[148] S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. P. Mailoa, M. Kornbluth, N. Molinari, T. E. Smidt, and B. Kozinsky, "E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials," *Nature Communications*, vol. 13, p. 2453, May 2022.

[149] J. Brehmer, S. Behrends, P. de Haan, and T. Cohen, "Does equivariance matter at scale?," 2025.

[150] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller, "SchNet – A deep learning architecture for molecules and materials," *The Journal of Chemical Physics*, vol. 148, p. 241722, June 2018.

[151] K. T. Schütt, P. Kessel, M. Gastegger, K. A. Nicoli, A. Tkatchenko, and K.-R. Müller, "SchNetPack: A Deep Learning Toolbox For Atomistic Systems," *Journal of Chemical Theory and Computation*, vol. 15, pp. 448–455, Jan. 2019.

[152] K. T. Schütt, S. S. P. Hessmann, N. W. A. Gebauer, J. Lederer, and M. Gastegger, "SchNetPack 2.0: A neural network toolbox for atomistic machine learning," *The Journal of Chemical Physics*, vol. 158, p. 144801, Apr. 2023.

[153] T. Xie and J. C. Grossman, "Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties," *Physical Review Letters*, vol. 120, p. 145301, Apr. 2018.

[154] V. G. Satorras, E. Hoogeboom, and M. Welling, "E(n) equivariant graph neural networks," 2022.

[155] J. T. M. Johannes Gasteiger, Shankari Giri and S. Günnemann, "Fast and uncertainty- aware directional message passing for non-equilibrium molecules. in machine learning for molecules workshop, neurips," 2020.

[156] J. Gasteiger, F. Becker, and S. Günnemann, "Gemnet: Universal directional graph neural networks for molecules," in *Advances in Neural Information Processing Systems* (M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, eds.), vol. 34, pp. 6790–6802, Curran Associates, Inc., 2021.

[157] J. Gasteiger, F. Becker, and S. Günnemann, "Gemnet: Universal directional graph neural networks for molecules," 2024.

[158] M. Weiler, M. Geiger, M. Welling, W. Boomsma, and T. Cohen, "3d steerable cnns: Learning rotationally equivariant features in volumetric data," 2018.

[159] R. Kondor, Z. Lin, and S. Trivedi, "Clebsch-gordan nets: a fully fourier space spherical convolutional neural network," 2018.

[160] I. Batatia, S. Batzner, D. P. Kovács, A. Musaelian, G. N. C. Simm, R. Drautz, C. Ortner, B. Kozinsky, and G. Csányi, "The design space of e(3)-equivariant atom-centered interatomic potentials," 2022.

[161] N. Thomas, T. Smidt, S. Kearnes, L. Yang, L. Li, K. Kohlhoff, and P. Riley, "Tensor field networks: Rotation- and translation-equivariant neural networks for 3d point clouds," 2018.

[162] A. Musaelian, S. Batzner, A. Johansson, L. Sun, C. J. Owen, M. Kornbluth, and B. Kozinsky, "Learning local equivariant representations for large-scale atomistic dynamics," *Nature Communications*, vol. 14, p. 579, Feb. 2023.

[163] A. Musaelian, A. Johansson, S. Batzner, and B. Kozinsky, "Scaling the leading accuracy of deep equivariant models to biomolecular simulations of realistic size," 2023.

[164] I. Batatia, D. P. Kovács, G. N. C. Simm, C. Ortner, and G. Csányi, "Mace: Higher order equivariant message passing neural networks for fast and accurate force fields," 2023.

[165] D. P. Kovács, I. Batatia, E. S. Arany, and G. Csányi, "Evaluation of the MACE force field architecture: From medicinal chemistry to materials science," *The Journal of Chemical Physics*, vol. 159, p. 044118, July 2023.

[166] D. P. Kovács, J. H. Moore, N. J. Browning, I. Batatia, J. T. Horton, Y. Pu, V. Kapil, W. C. Witt, I.-B. Magdău, D. J. Cole, and G. Csányi, "MACE-OFF: Short-Range Transferable Machine Learning Force Fields for Organic Molecules," *Journal of the American Chemical Society*, vol. 147, pp. 17598–17611, May 2025.

[167] B. Mortazavi, "Recent advances in machine learning-assisted multiscale design of energy materials," *Advanced Energy Materials*, vol. 15, no. 9, p. 2403876, 2025.

[168] Y. Park, J. Kim, S. Hwang, and S. Han, "Scalable parallel algorithm for graph neural network interatomic potentials in molecular dynamics simulations," *Journal of Chemical Theory and Computation*, vol. 20, p. 4857–4868, May 2024.

[169] J. Kim, J. Kim, J. Kim, J. Lee, Y. Park, Y. Kang, and S. Han, "Data-Efficient Multifidelity Training for High-Fidelity Machine Learning Interatomic Potentials," *Journal of the American Chemical Society*, vol. 147, pp. 1042–1054, Jan. 2025.

[170] M. Neumann, J. Gin, B. Rhodes, S. Bennett, Z. Li, H. Choubisa, A. Hussey, and J. Godwin, "Orb: A fast, scalable neural network potential," 2024.

[171] B. Rhodes, S. Vandenhaute, V. Šimkus, J. Gin, J. Godwin, T. Duignan, and M. Neumann, "Orb-v3: atomistic simulation at scale," 2025.

[172] C. Chen and S. P. Ong, "A universal graph deep learning interatomic potential for the periodic table," *Nature Computational Science*, vol. 2, pp. 718–728, Nov. 2022.

[173] B. Deng, P. Zhong, K. Jun, J. Riebesell, K. Han, C. J. Bartel, and G. Ceder, "CHGNet as a pretrained universal neural network potential for charge-informed atomistic modelling," *Nature Machine Intelligence*, vol. 5, pp. 1031–1041, Sept. 2023.

[174] A. Merchant, S. Batzner, S. S. Schoenholz, M. Aykol, G. Cheon, and E. D. Cubuk, "Scaling deep learning for materials discovery," *Nature*, vol. 624, pp. 80–85, Dec. 2023.

[175] I. Batatia, S. Batzner, D. P. Kovács, A. Musaelian, G. N. C. Simm, R. Drautz, C. Ortner, B. Kozinsky, and G. Csányi, "The design space of E(3)-equivariant atom-centred interatomic potentials," *Nature Machine Intelligence*, vol. 7, pp. 56–67, Jan. 2025.

[176] I. Batatia and ..., "A foundation model for atomistic materials chemistry," *The Journal of Chemical Physics*, vol. 163, p. 184110, Nov. 2025.

[177] M. Radova, W. G. Stark, C. S. Allen, R. J. Maurer, and A. P. Bartók, "Fine-tuning foundation models of materials interatomic potentials with frozen transfer learning," *npj Computational Materials*, vol. 11, p. 237, July 2025.

[178] https://docs.deepmodeling.com/projects/deepmd/en/latest/troubleshooting/howtoset_netsize.html.

[179] A. Fantasia, F. Rovaris, O. Abou El Kheir, A. Marzegalli, D. Lanzoni, L. Pessina, P. Xiao, C. Zhou, L. Li, G. Henkelman, E. Scalise, and F. Montalenti, "Development of a machine learning interatomic potential for exploring pressure-dependent kinetics of phase transitions in germanium," *The Journal of Chemical Physics*, vol. 161, p. 014110, July 2024.

[180] E. V. Podryabinkin and A. V. Shapeev, "Active learning of linearly parametrized interatomic potentials," *Computational Materials Science*, vol. 140, pp. 171–180, Dec. 2017.

[181] J. S. Smith, B. Nebgen, N. Lubbers, O. Isayev, and A. E. Roitberg, "Less is more: Sampling chemical space with active learning," *The Journal of Chemical Physics*, vol. 148, p. 241733, June 2018.

[182] K. Gubaev, E. V. Podryabinkin, G. L. Hart, and A. V. Shapeev, "Accelerating high-throughput searches for new alloys with active

learning of interatomic potentials," *Computational Materials Science*, vol. 156, pp. 148–156, Jan. 2019.

[183] C. Schran, K. Brezina, and O. Marsalek, "Committee neural network potentials control generalization errors and enable active learning," *The Journal of Chemical Physics*, vol. 153, p. 104105, Sept. 2020.

[184] A. Jain, S. P. Ong, Geoffroy Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, and K. A. Persson, "Commentary: The Materials Project: A materials genome approach to accelerating materials innovation," *APL Materials*, vol. 1, p. 011002, July 2013.

[185] A. Mujica, C. J. Pickard, and R. J. Needs, "Low-energy tetrahedral polymorphs of carbon, silicon, and germanium," *Physical Review B*, vol. 91, p. 214104, June 2015.

[186] Z. Zhao, F. Tian, X. Dong, Q. Li, Q. Wang, H. Wang, X. Zhong, B. Xu, D. Yu, J. He, H.-T. Wang, Y. Ma, and Y. Tian, "Tetragonal Allotrope of Group 14 Elements," *Journal of the American Chemical Society*, vol. 134, pp. 12362–12365, Aug. 2012.

[187] A. Ghasemi, P. Xiao, and W. Gao, "Nudged elastic band method for solid-solid transition under finite deformation," *The Journal of Chemical Physics*, vol. 151, p. 054110, Aug. 2019.

[188] J.-T. Wang, C. Chen, H. Mizuseki, and Y. Kawazoe, "Kinetic origin of divergent decompression pathways in silicon and germanium," *Physical Review Letters*, vol. 110, p. 165503, Apr. 2013.

[189] F. Rovaris, A. Marzegalli, F. Montalenti, and E. Scalise, "Unraveling the atomic-scale pathways driving pressure-induced phase transitions in silicon," *Materials Today Nano*, vol. 29, p. 100548, 2025.

[190] A. Stukowski, "Visualization and analysis of atomistic simulation data with OVITO–the Open Visualization Tool," *Modelling and Simulation in Materials Science and Engineering*, vol. 18, p. 015012, Jan. 2010.

[191] A. Fantasia et al., "A NN potential for phase transformations in Ge." Materials Cloud Archive 2024.55; https://doi.org/10.24435/materialscloud:r2-qc.

[192] Y. Kok, X. P. Tan, P. Wang, M. Nai, N. H. Loh, E. Liu, and S. B. Tor, "Anisotropy and heterogeneity of microstructure and mechanical properties in metal additive manufacturing: A critical review," *Materials & Design*, vol. 139, pp. 565–586, 2018.

[193] X. Li, L. Lu, J. Li, X. Zhang, and H. Gao, "Mechanical properties and deformation mechanisms of gradient nanostructured metals and alloys," *Nature Reviews Materials*, vol. 5, pp. 706–723, 2020.

[194] C. Hu, S. Martin, and R. Dingreville, "Accelerating phase-field predictions via recurrent neural networks learning the microstructure evolution in latent space," *Computer Methods in Applied Mechanics and Engineering*, vol. 397, p. 115128, July 2022.

[195] O. Ahmad, R. Maurya, R. Mukherjee, and S. Bhowmick, "Integrated Phase Field and Machine Learning Study of Microstructure Evolution during Interface-Controlled Spinodal Decomposition," *SSP*, vol. 357, pp. 101–106, June 2024.

[196] P. Wu, A. S. Iquebal, and K. Ankit, "Emulating microstructural evolution during spinodal decomposition using a tensor decomposed convolutional and recurrent neural network," *Computational Materials Science*, vol. 224, p. 112187, May 2023.

[197] D. Lanzoni, F. Rovaris, L. Martín-Encinar, A. Fantasia, R. Bergamaschini, and F. Montalenti, "Accelerating simulations of strained-film growth by deep learning: Finite element method accuracy over long time scales," *APL Machine Learning*, vol. 2, p. 036108, 2024.

[198] J. Y. Choi, T. Xue, S. Liao, and J. Cao, "Accelerating phase-field simulation of three-dimensional microstructure evolution in laser powder bed fusion with composable machine learning predictions," *Additive Manufacturing*, vol. 79, p. 103938, Jan. 2024.

[199] W. Soffa and D. E. Laughlin, "8 - diffusional phase transformations in the solid state," in *Physical Metallurgy (Fifth Edition)* (D. E. Laughlin and K. Hono, eds.), pp. 851–1020, Oxford: Elsevier, fifth edition ed., 2014.

[200] S. Brenner, P. Camus, M. Miller, and W. Soffa, "Phase separation and coarsening in fecrco alloys," *Acta Metallurgica*, vol. 32, no. 8, pp. 1217–1227, 1984.

[201] H. Garcke, S. Maier-Paape, and U. Weikard, "Spinodal Decomposition in the Presence of Elastic Interactions," in *Geometric Analysis and Nonlinear Partial Differential Equations* (S. Hildebrandt and H. Karcher, eds.), pp. 603–635, Berlin, Heidelberg: Springer, 2003.

[202] T. Mura, *Micromechanics of Defects in Solids*, vol. 3 of *Mechanics of Elastic and Inelastic Solids*. Dordrecht: Springer Netherlands, 1987.

[203] A. Khachaturyan, *Theory of Structural Transformations in Solids*. Dover Books on Engineering Series, Dover Publications, 2008.

[204] J. W. Cahn, "On spinodal decomposition in cubic crystals," *Acta Metallurgica*, vol. 10, pp. 179–183, 1962.

[205] J. Zhu, L.-Q. Chen, and J. Shen, "Morphological evolution during phase separation and coarsening with strong inhomogeneous elasticity," *Modelling and Simulation in Materials Science and Engineering*, vol. 9, p. 499, 2001.

[206] E. Butler and G. Thomas, "Structure and properties of spinodally decomposed cu-ni-fe alloys," *Acta Metallurgica*, vol. 18, no. 3, pp. 347–365, 1970.

[207] Y. Wang, L.-Q. Chen, and A. Khachaturyan, "Kinetics of strain-induced morphological transformation in cubic alloys with a miscibility gap," *Acta Metallurgica et Materialia*, vol. 41, pp. 279–296, Jan. 1993.

[208] P. Fratzl, O. Penrose, and J. L. Lebowitz, "Modeling of Phase Separation in Alloys with Coherent Elastic Misfit," *Journal of Statistical Physics*, vol. 95, no. 5/6, pp. 1429–1503, 1999.

[209] K. Perlin, "An image synthesizer," *SIGGRAPH Comput. Graph.*, vol. 19, pp. 287–296, July 1985.

[210] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, 1998.

[211] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, pp. 1735–1780, 11 1997.

[212] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," 2014.

[213] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," in *A Field Guide to Dynamical Recurrent Neural Networks* (S. C. Kremer and J. F. Kolen, eds.), IEEE Press, 2001.

[214] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proceedings of the 30th International Conference on Machine Learning* (S. Dasgupta and D. McAllester, eds.), vol. 28 of *Proceedings of Machine Learning Research*, (Atlanta, Georgia, USA), pp. 1310–1318, PMLR, 17–19 Jun 2013.

[215] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," 2013.

[216] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in *Proceedings of the 32nd International Conference on Machine Learning* (F. Bach and D. Blei, eds.), vol. 37 of *Proceedings of Machine Learning Research*, (Lille, France), pp. 2342–2350, PMLR, 07–09 Jul 2015.

[217] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," May 2015. arXiv:1505.04597 [cs].

[218] A. Kamath, J. Willmann, N. Andratschke, and M. Reyes, "The impact of u-net architecture choices and skip connections on the robustness of segmentation across texture variations," *Computers in Biology and Medicine*, vol. 197, p. 111056, 2025.

[219] T. Chen and H. Chen, "Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems," *IEEE Transactions on Neural Networks*, vol. 6, no. 4, pp. 911–917, 1995.

[220] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, "Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators," *Nature Machine Intelligence*, vol. 3, pp. 218–229, Mar. 2021.

[221] W. Li, M. Z. Bazant, and J. Zhu, "Phase-field deeponet: Physics-informed deep operator neural network for fast simulations of pattern formation governed by gradient flows of free-energy functionals," 2023.

[222] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, "Fourier neural operator for parametric partial differential equations," 2021.

[223] E. M. Minete, M. Immertreu, F. Teichmann, and S. Müller, "Fourier neural operators for two-phase, 2d mold-filling problems related to metal casting," 2025.

[224] C. Bonneville, N. Bieberdorf, A. Hegde, M. Asta, H. N. Najm, L. Capolungo, and C. Safta, "Accelerating phase field simulations through a hybrid adaptive Fourier neural operator with U-net backbone," *npj Computational Materials*, vol. 11, p. 14, Jan. 2025.

[225] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

[226] S. K. Boya and D. N. Subramani, "Pinto: Physics-informed transformer neural operator for learning generalized solutions of partial differential equations for any initial and boundary condition," *Computer Physics Communications*, vol. 315, p. 109702, 2025.

[227] D. Ruhe, J. Heek, T. Salimans, and E. Hoogeboom, "Rolling diffusion models," 2024.

[228] D. Lanzoni, M. Albani, R. Bergamaschini, and F. Montalenti, "Morphological evolution via surface diffusion learned by convolutional, recurrent neural networks: Extrapolation and prediction uncertainty," *Phys. Rev. Materials*, vol. 6, p. 103801, Oct. 2022.

[229] S. Schubert, P. Neubert, J. Poschmann, and P. Protzel, "Circular Convolutional Neural Networks for Panoramic Images and Laser Data," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, (Paris, France), pp. 653–660, IEEE, 2019.

[230] M. Salvalaglio, R. Backofen, R. Bergamaschini, F. Montalenti, and A. Voigt, "Faceting of Equilibrium and Metastable Nanostructures: A Phase-Field Model of Surface Diffusion Tackling Realistic Shapes," *Crystal Growth & Design*, vol. 15, pp. 2787–2794, 2015.

[231] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th Annual International Conference on Machine Learning*, (Montreal Quebec Canada), pp. 41–48, ACM, June 2009.

[232] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, (San Diego, CA, USA), pp. 886–893, IEEE, 2005.

[233] N. Sarkar and B. Chaudhuri, "An efficient differential box-counting approach to compute fractal dimension of image," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, pp. 115–120, Jan. 1994.

[234] `https://doi.org/10.24435/materialscloud:b4-9t`, reference number 2025.117, 2025.

[235] `https://github.com/dlanzo/CRANE`, 2025.

[236] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, "Fourier Neural Operator for Parametric Partial Differential Equations," May 2021. arXiv:2010.08895 [cs].