

# Cluster Expansion Monte-Carlo simulations of InGaN alloys

Thomas Sidos



Department of Physics

University of Crete

June 2025

## Abstract

The  $\text{In}_x\text{Ga}_{1-x}\text{N}$  alloy system plays a central role in modern optoelectronics, enabling high-efficiency light-emitting diodes (LEDs), laser diodes, and solar cells. Nevertheless, understanding the thermodynamics of bulk InGaN alloys is therefore essential for controlling phase stability, atomic ordering, and growth processes. In this thesis, we present a comprehensive investigation of the bulk alloy thermodynamics of  $\text{In}_x\text{Ga}_{1-x}\text{N}$  using a first-principles-based cluster expansion (CE) model combined with Monte Carlo (MC) simulations. The study spans the full compositional range ( $x = 0.05\text{--}0.95$ ) and temperatures from 0 K to 1800 K, focusing on pseudomorphic alloys coherently strained to GaN substrates.

Our calculations reveal that, the zero-temperature mixing energy landscape exhibits a single global minimum, indicating the suppression of decomposition into pure GaN and InN under biaxial strain. Thermodynamic properties computed via MC simulations reveal a rich variety of configurational phases. A temperature-composition phase diagram highlights the emergence of order–disorder transitions in the 600–750 K range for indium concentrations between 20–40%.

Microscopic analysis using Warren–Cowley short-range order parameters uncovers pronounced strain-induced ordering motifs at low temperatures, such as  $\sqrt{3}\times\sqrt{3}$ -type in-plane patterns and second-neighbor vertical stacking, especially near  $x = 0.25\text{--}0.35$ . These motifs are gradually disrupted near the critical temperature ( $T_c$ ), above which the alloy transitions into a disordered, entropy-driven phase. The order–disorder transitions are corroborated by sharp peaks in the heat capacity and the loss of SRO correlations.

Interestingly, while full phase separation into GaN and InN is thermodynamically disfavored under coherent strain, our results indicate a local instabilities, where the alloy exhibits a weak tendency to separate into compositions richer and poorer in indium. These tendencies manifest at relatively low temperatures and remain a subject for future experimental verification.

Overall, this work underscores the critical role of alloy thermodynamics in guiding the synthesis and optimization of InGaN-based devices. The detailed thermodynamic and microscopic insights provided here lay the groundwork for better predictive control of alloy composition, phase stability, and atomic-scale ordering in strained nitride heterostructures.

---

## Acknowledgments

Above all, I would like to express my appreciation to my advisor Mr. L. Lymperakis for his unique and remarkable patience as well as for his guidance and support during the construction of this thesis. He has granted me the opportunities to focus and improve my efforts in understanding and doing this work.

Also, I deeply thank my family for their confidence and encouragement in equal measure. Their all round support, emotional as well as practical, has truly formed an anchor in my education.

To all my friends who stood with my side during this challenge and in my success, I thank you sincerely. Your presence is not only a walk but at times a run and to great delight all through this journey.

Also, I wish to express my appreciation for the services of my university that in one way or the other have supported the achievement of research objectives and goals. The libraries, laboratories, and other academic resources available to me have indeed assisted my work greatly.

Lastly, I would like to thank everyone for their broad support, irrespective of how much effort they put in, to enable me accomplish this thesis.

---

## Contents

<b>1 Introduction</b>	<b>1</b>
1.1 III-Nitrides.....	2
1.2 $\text{In}_x\text{Ga}_{1-x}\text{N}$ .....	3
1.3 Crystal Structure of InGaN.....	4
1.4 Alloy Cluster Expansion.....	5
1.4.1 ICET CE Python Package.....	7
1.5 Monte Carlo Simulations.....	11
1.6 Warren-Cowley Short-Range Order (SRO) Parameter.....	14
1.7 Spinodal Decomposition.....	17
1.8 Phase Diagram Construction: Chemical Potentials and Spinodal Curves.....	18
<b>2 Results and Discussion</b>	<b>21</b>
2.1 Cluster Expansion Results.....	21
2.2 Monte Carlo Calculations.....	24
2.3 Phase Diagram of $\text{In}_x\text{Ga}_{1-x}\text{N}$ alloy.....	26
2.4 Configurational Properties.....	30

---

<b>3 Conclusions</b>	<b>42</b>
<b>Appendix</b>	<b>44</b>
Appendix A : Computational Scripts	
A.0 DFT Data Processing to Database.....	44
A.1 Cluster Cutoff Optimization.....	47
A.2 CE Fitting and Validation.....	50
A.3 MC Simulation of Thermodynamic Properties.....	53
A.4 Phase Diagram-Chemical Potential Difference $\Delta\mu$ .....	57
A.5 Thermodynamic Analysis-Entropy and Free Energy of Mixing.....	59
A.6 Warren-Cowley SRO Parameter Analysis.....	62
Appendix B : Regression Methods	66
B1 LASSO Regression.....	66
B2 Bayesian Regression.....	66
Appendix C : Effective Cluster Interactions (ECIs)	66
<b>References</b>	<b>68</b>

# Chapter 1

## Introduction

To study the design, growth and composition of an alloy, we must first obtain thermodynamic information about these materials and, in particular, derive and understand the corresponding phase diagrams.

These diagrams contain information about the stability, phase transitions and general behavior of the material under conditions such as temperature, pressure and composition.

Another key factor that arises from the construction of the phase diagram is the atomic configurations in the microscopic scale which determine the properties of alloys at a macroscopic level. These configurations can significantly impact the optoelectronic properties of semiconducting materials or the mechanical properties of metallic structural materials.

As been described in the abstract, in Computational Materials , first principles calculations and methods such as Cluster Expansion (CE) and Monte-Carlo (MC) are combined so to derive the properties mentioned before. The procedure here is that a CE Hamiltonian is trained and validated in respect of Density Functional Theory (DFT) calculations. The Hamiltonian efficiently and accurately describes the energy configuration of alloy systems being comprised of thousand, even millions of atoms. This efficient energy configuration is used to perform Monte-Carlo simulations. The calculations, can be done in various thermodynamic ensembles such as grandcanonical, canonical, etc. , contain valuable information including chemical potentials, total/mixing energies and heat capacities as functions of temperature. This precious data can be deployed to derive phase diagrams and then observe order-disorder transitions within the system.

Another important outcome of these calculations is the generation of atomic configurations at various temperatures and alloy compositions. These configurations provide valuable information about the structural evolution of the material against varying thermodynamic conditions such as order-disorder phenomena.

## 1.1 III-Nitrides

Group-III nitrides, which include Aluminium Nitride (AlN), Gallium Nitride (GaN), and Indium Nitride (InN), form an important class of semiconductors. Since the 1970s, researchers have recognized their potential for applications in electronics and optoelectronics[2]. A major breakthrough occurred in the early to mid-1990s with the development of the first highly efficient blue light-emitting diodes (LEDs) and laser diodes (LDs)[7]. This progress opened the door to numerous optical technologies, such as violet, blue, and green LEDs, as well as high-power and high-frequency transistors.

These materials consist of nitrogen combined with one or more group-III elements. They are characterized by strong atomic bonds, high melting points, and excellent thermal conductivity. Furthermore, III-nitride semiconductors exhibit direct bandgaps

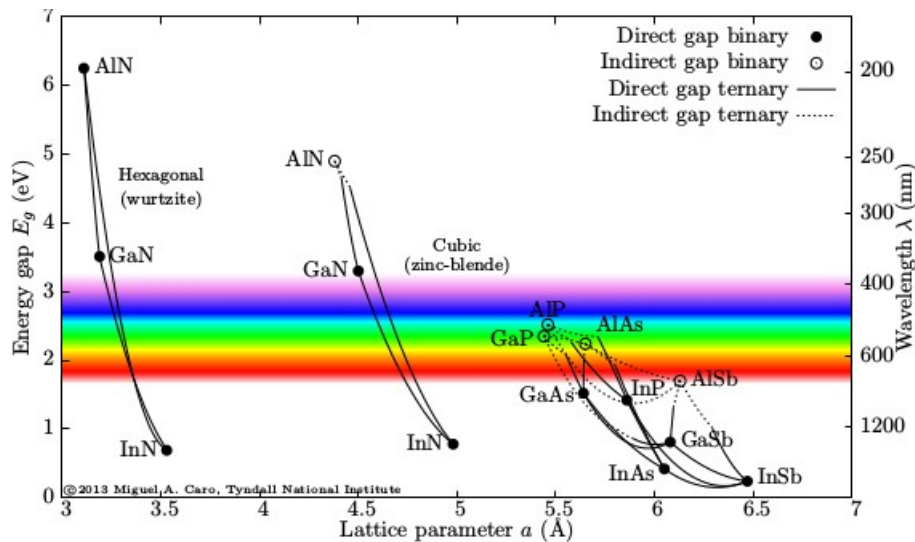


Figure 1.1 :Bandgaps of selected semiconductors as a function of lattice constant [23]

spanning from the ultraviolet (UV) to the infrared (IR) spectrum(see Fig.1.1). This direct bandgap property facilitates efficient radiative recombination of electrons, a key factor in the performance of optoelectronic devices.

One of the key advantages of these materials is the ability to engineer their bandgap by varying the composition of alloys, such as  $\text{In}_x\text{Ga}_{1-x}\text{N}$ . This tunability is promising for the creation of a range of optoelectronic devices that emit light in different wavelengths, such as UV and visible LEDs, infrared photodetectors, and multi-wavelength laser diodes.

### 1.2 $\text{In}_x\text{Ga}_{1-x}\text{N}$

$\text{In}_x\text{Ga}_{1-x}\text{N}$  is a ternary semiconductor alloy composed of indium nitride (InN) and gallium nitride (GaN), with tunable optical and electronic properties depending on the indium composition ( $x$ ). By varying  $x$ , the bandgap energy can be engineered from approximately 0.7 eV (InN) to 3.4 eV (GaN), enabling applications across a wide spectral range from infrared to ultraviolet. First-principles calculations and experimental studies have shown that increasing indium content results in lattice expansion, alter the material's electronic band structure and influences charge carrier dynamics. The wurtzite crystal structure remains stable across a range of compositions, though strain-induced defects such as dislocations play a crucial role in lattice relaxation.

This material has garnered significant interest in optoelectronics, particularly for high-efficiency light-emitting diodes (LEDs), laser diodes, and high-performance solar cells. The tunability of its bandgap allows for multi-junction solar cells and high-efficiency photodetectors. Additionally, nanostructured forms such as  $\text{In}_x\text{Ga}_{1-x}\text{N}$  nanorods and quantum wells exhibit enhanced optical properties due to reduced strain and quantum confinement effects, further broadening its potential applications. Advancements in III-Nitrides growth, including molecular beam epitaxy (MBE) and metal-organic chemical vapor deposition (MOCVD), have improved compositional control and defect reduction, making  $\text{In}_x\text{Ga}_{1-x}\text{N}$  the material of choice for next-generation optoelectronic devices.

A fundamental challenge in the synthesis of  $\text{In}_x\text{Ga}_{1-x}\text{N}$  quantum wells is the difficulty in incorporating indium beyond a certain threshold while maintaining coherent growth. Lymperakis et al. [6] identified a compositional limit of  $\sim 25\%$  In under N-rich growth on GaN(0001), attributing it to a novel mechanism they termed elastically frustrated rehybridization. This mechanism explains the experimentally observed ordered configurations and the inhibition of further In incorporation despite variations in growth conditions.

Recent ab initio studies have shown that strain has a significant impact on the thermodynamic stability of  $\text{In}_x\text{Ga}_{1-x}\text{N}$  alloys. Duff et al. [21] demonstrated that coherent epitaxial growth suppresses spinodal decomposition by increasing the total energy of strained InN. As a result, the Gibbs free energy of mixing remains negative with a single minimum across the full composition range, eliminating the thermodynamic driving force for phase separation towards GaN and InN.

This thesis investigates the configurational and thermodynamic properties of InGaN alloys using computational material's methodologies. MC simulations and DFT calculations are employed to analyze the atomic arrangements and phase stability of InGaN over a range of compositions and temperatures. Special attention is given to the ordering phenomena at the nanoscale.



### 1.3 Crystal structure of InGaN

This section focuses on the structure of  $\text{In}_x\text{Ga}_{1-x}\text{N}$ . Both InN and GaN crystallize in the wurtzite structure.  $\text{In}_x\text{Ga}_{1-x}\text{N}$  alloys also adopt the hexagonal wurtzite structure.[10]

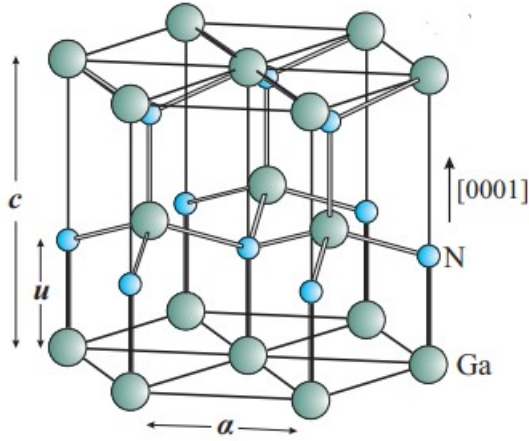


Figure 1.3 : Schematic representation in ball and stick model of the GaN wz structure. N denotes the Nitrogen atoms and Ga the Gallium atoms. [10]

This structure can be described by two sublattices, the metal and the Nitrogen sublattice. The primitive vectors that describe the translational symmetry are the following:

$$a_1 = \left[ \frac{1}{2}a, -\frac{\sqrt{3}}{2}a, 0 \right] \quad a_2 = \left[ \frac{1}{2}a, \frac{\sqrt{3}}{2}a, 0 \right] \quad a_3 = [0, 0, c] \quad (1.1)$$

where  $a$  and  $c$  are lattice constants. In the ideal wurtzite crystal  $c = \sqrt{\frac{8}{3}}a$

Crystal points are at positions (direct/reduced coordinates) :

$$\begin{aligned} b_1 &= [0, 0, 0] \\ b_2 &= \left[ \frac{2}{3}, \frac{1}{3}, \frac{1}{2} \right] \\ b_3 &= [0, 0, u] \\ b_4 &= \left[ \frac{2}{3}, \frac{1}{3}, u+0,5 \right] \end{aligned} \quad (1.2)$$

In this representation  $u$  is an internal lattice parameter which describes the shift of the one sublattice in respect to the other. In the ideal wurtzite structure  $u = \frac{3}{8}$ . Vectors  $b_i$  are the basis vectors.

Antisite point defects, denoted as  $M_N$  and  $N_M$ , occur when metal atoms ( $M$ :In,Ga or Al) occupy positions intended for non-metal atoms ( $N$ ), and vice-versa. These defects are characterized by the exchange of atom positions between metal and non-metal elements within the crystal structure. The formation of antisite defects generally requires high formation energies, making them relatively rare. Therefore, their concentrations are typically very small. In consequence we can approximate our InGa<sub>N</sub> alloy as a pseudobinary  $In_xGa_{1-x}$  alloy with hexagonal symmetry and so the primitive vectors are the same but we keep only the  $b_1$  and  $b_2$ , basis vectors, i.e., we treat the metal sublattice only. It is now clear that our primitive cell of our system is described by the primitive vectors of (1.1) and the two basis vectors of (1.2)

In addition, the volume of the primitive cell is given by the following equation:

$$V = |a_1 \cdot (a_2 \times a_3)| \quad (1.3)$$

where  $a_1$ ,  $a_2$ ,  $a_3$  are the primitive vectors.

### 1.4 Alloy Cluster Expansion

Energetics of alloy systems are well-established not only for the usage in the study of multi-component materials, but also to predict phase stability, phase transitions and chemical ordering. However, for large configurational spaces, direct first-principles calculations become computationally prohibitive. This limitation can be addressed in a systematic and efficient manner using the cluster expansion (CE) method, which maps the configurational dependence of a property of interest (most commonly the total energy) onto a discrete lattice model.

The total energy of a material is expressed using the cluster expansion formalism as the sum of the contributions from each individual atomic site and their interactions within atom clusters. Weighted by effective cluster interaction (ECI) coefficients, these clusters comprise pairs, triplets, and higher-order interactions. The method, in principle, maintains high accuracy with low computational cost

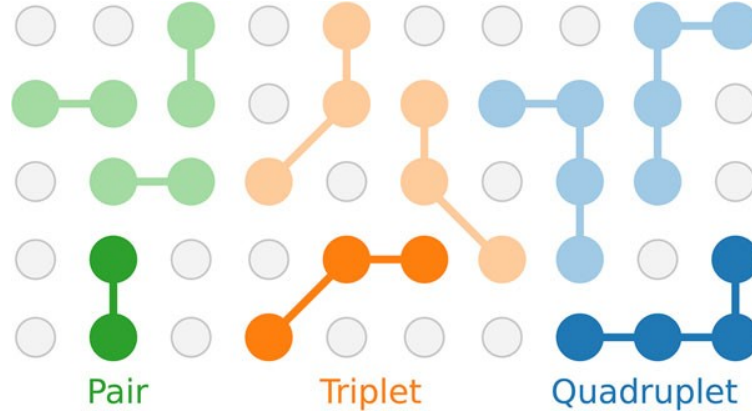


Figure 1.4 : Examples for pair,triplet and quadruplet clusters on a square lattice. The representative (symmetry inequivalent) clusters are shown in dark colors, whereas examples for other clusters in the orbits are shaded.[25]

At its core, the cluster expansion expresses a target property, such as the total energy  $E(\sigma)$ , as a series expansion in terms of occupation variables that describe the distribution of atomic species on a lattice. This expansion is constructed using a complete orthonormal basis set, where the energy contribution from different atomic configurations is decomposed into a sum over clusters of sites:

$$E(\sigma) = E_0 + \sum_{\alpha} J_{\alpha} \langle \Phi_{\alpha}(\sigma) \rangle \quad (1.4)$$

where :

$E_0$  : Is the reference ,

$J_{\alpha}$  : Represent the effective cluster interaction (ECI) coefficients that quantify the contribution of each cluster  $\alpha$  , and

$\Phi_{\alpha}(\sigma)$  : Is the cluster basis functions that encode the configurational dependence of the energy .

The occupation variables  $\sigma_i$  take discrete values depending on the atomic species occupying site  $i$  .

$$\sigma_i = \begin{cases} +1 & \text{if site } i \text{ is occupied by In} \\ -1 & \text{if site } i \text{ is occupied by Ga} \end{cases} \quad (1.5)$$

To construct the CE, clusters are defined as groups of sites interacting with each other. Each cluster  $\alpha$  contributes to the total energy with a weight determined by its corresponding basis function  $\Phi_{\alpha}(\sigma)$ , which can be expressed as a product of occupation variables:

$$\Phi_{\alpha}(\sigma) = \prod_{i \in \alpha} \sigma_i \quad (1.6)$$

The expansion includes interactions of different orders :

- Single terms ( $\sigma_i$ ) : On-site energy contribution
- Pair interactions ( $\sigma_i \sigma_j$ ) : Nearest-neighbor or longer-range pairwise interactions
- Triplet and higher order interactions : Account for multi-body simple interactions

The ECI's  $J_{\alpha}$  are determined by fitting the CE to a set of reference energies obtained from first-principles calculations (see Appendix C). This results in a system of linear equations:

$$E_k = E_0 + \sum_{\alpha} J_{\alpha} \langle \Phi_{\alpha}(\sigma_k) \rangle \quad (1.7)$$

where  $E_0$  are the DFT-calculated energies for different atomic configurations . The solution for can be obtained using statistical regression techniques (Least Squares Regression, LASSO, Bayesian Regression) (see Appendix 2.1)

So, by doing all that, the Hamiltonian that describes our system, is built. Therefore MC calculations can be done by implementing CE Hamiltonian.

### 1.4.1 ICET CE Python Package

The python library for fitting a CE Hamiltonian and performing MC calculations is ICET. ICET is largely written in Python for easy integration in comprehensive workflows, including first-principles calculations for the generation of reference data and machine learning libraries for training and validation. For our CE and MC calculations we used the ICET library [citation]

### ICET Cluster Expansion Formalism :

The CE formalism provides a systematic and efficient method to represent the energy of a crystalline system as a function of atomic configurations. This approach is particularly useful in

alloy theory and computational materials science, where first-principles calculations are computationally prohibitive to describe for large configuration spaces. The CE formalism decomposes the total energy of a system into contributions from clusters of atoms.

The formalism here is the typical CE formalism mentioned before. **Although a discussion about Machine Learning (ML) in python must be made.** Since the number of possible clusters grows exponentially with system size, a practical cluster expansion includes only the most significant interaction terms. Machine learning approaches, such as LASSO regression and Bayesian model selection, provide automated methods to determine which clusters contribute most significantly.

Traditionally, ECIs  $J_\alpha$  are obtained by fitting the cluster expansion model to a set of first-principles calculations. Machine learning improves this process by using advanced regression techniques mentioned before.

The optimization process typically minimizes a cost function:

$$\sum_n (E_n^{DFT} - E(\sigma_n))^2 + \lambda \sum_\alpha |J_\alpha| \quad (1.8)$$

where  $E_n^{DFT}$  is the DFT-calculated energy of configuration , and  $\lambda$  is a regularization parameter to control model complexity. In practice,  $\lambda$  is typically determined through cross-validation or selected based on prior experience to achieve optimal performance.

The performance of different CE models can be assessed by cross-validation (CV). To this end, the available reference data is split into training and validation sets. The former is used to fit a model, whereas the latter is used to measure the predictive power of the model, usually via the root-mean-square error (RMSE):

$$RMSE = \sqrt{\frac{1}{N_s} \sum_i (Q_i^{model} - Q_i^{target})^2} \quad (1.9)$$

where the summation extends over the  $N_s$  structures comprising the validation set. To reduce the statistical error, the RMSE is furthermore averaged over several different splits of the reference data.

For a better understanding, the workflow of CE will be analyzed below using a matrix form

formalism. First ECI's must be determined, a set of reference data is required as a set of configurations  $\{\sigma_1, \sigma_2, \dots, \sigma_n\}$  as well as an associated vector of target data  $Q^T = [Q_1, Q_2, \dots, Q_n]$  that can be obtained from first-principles calculations (DFT's). So, Eq. (1.7) can be transformed in this form :

$$Q = \Pi J \quad (1.10)$$

where the rows of  $\Pi$  are given by :

$$\Pi_i = [1, \langle \Pi_{\alpha_1}(\sigma_i) \rangle_{\alpha_1} m_{a_1}, \dots, \langle \Pi_{\alpha_n}(\sigma_i) \rangle_{\alpha_n} m_{a_n}] \quad (1.11)$$

$\Pi$  is the same quantity mentioned before in Eq (1.6) and  $J$  in Eq (1.10) denotes the ECI's with  $J_0=Q_0$ . The  $E_k$  part of Eq (1.7) is now denoted as  $Q$  in the matrix form.

Following the decomposition of a lattice into clusters, it remains to determine the ECI's by solving the linear system given by Eq (1.10). This is equivalent to finding the parameter vector  $J$  that minimizes  $\|\Pi J - Q\|^2$ . In principle a solution can be determined by conventional least-squares but there are more methods for linear regression such as: LASSO and Bayesian Regression. In this work we used the Bayesian Regression.

The workflow of our CE procedure is schematically shown in Fig. (1.5).

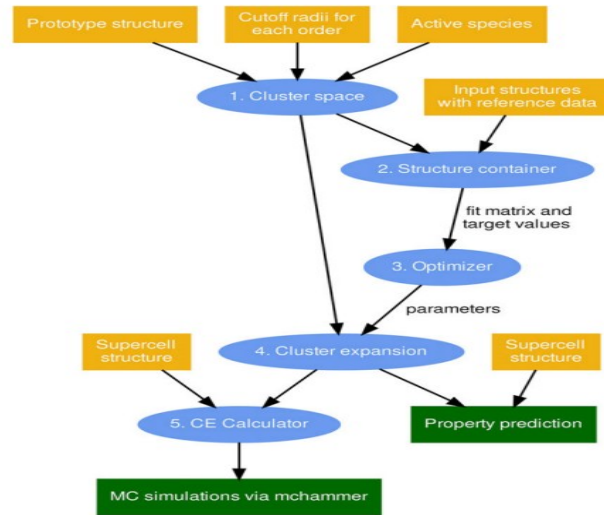


Figure 1.5 : Representation of the ICET workflow. Entities represented by Python objects are shown in blue, input parameters and data in orange and additional functionalities in green.[25]

First, a cluster space is defined by providing a prototype structure (primitive cell), the species allowed on each side (here In or Ga), and a set of cutoffs for each cluster order to be included. The values of cutoff radii's then specify the longest distance allowed between any two atoms in clusters of the respective order. Hence, the cluster expansions accuracy changes with cutoffs.

In ICET, the ClusterSpace object generates a symmetry-reduced set of clusters from a given parent lattice and cutoff radius. Each cluster is defined not only by its type (e.g., pair, triplet, e.t.c) but also by its cluster vector, which specifies the relative positions of the sites within the cluster. These vectors determine the spatial geometry of the interaction and are used internally to compute cluster functions and correlation vectors. For instance, a pair cluster might include all symmetry-

equivalent nearest-neighbor vectors within the cutoff. The cluster vectors are essential for encoding physical interactions and preserving lattice symmetry during the expansion.

Then, a structure container is initialized by using the cluster space created previously and a set of DFT input structures with reference data for the properties of interest (e.g. Mixing Energy). Basically a structure container is a collection of structures along with their cluster vectors. Structure containers allow one to easily compile structures for training and validation.

Since the object created in the previous section, contains all the information required for constructing a cluster expansion, the next step is to train the parameters, i.e. to fit the ECIs using the target mixing energies. More precisely, the goal is to achieve the best possible agreement with a set of training structures, which represent a subset of all the structures in the structure container. This task is achieved by the trainstation package (included in ICET).

At this point, the task of constructing the cluster expansion is nearly complete. The remaining step is to assign the optimized effective cluster interactions (ECIs) — the parameters obtained from the regression — to their corresponding clusters in the. In ICET, this is handled by creating a ClusterExpansion object, which maps each fitted coefficient to a specific cluster defined by its type and cluster vector. This association enables energy predictions for new configurations using the trained CE model. CEs are the main output of the ICET model construction cycle. While they are specific for a given prototype structure and cluster space they are not tied to a specific supercell structure.

The final cluster expansion can be used in a number of ways. Most commonly one creates a cluster expansion calculator for a specific supercell structure and subsequently carries out Monte Carlo simulations via the mchammer module.

### **Convex Hull Construction in ICET**

In addition to enabling cluster expansion fitting and Monte Carlo simulations, ICET also provides tools to analyze phase stability at zero temperature ( $T=0$  K) through the construction of the convex hull of formation energies.

The convex hull is a key thermodynamic concept in alloy theory: It is the lower envelope of the formation energy vs. composition plot, connecting the lowest-energy configurations at each composition. Configurations that lie on this hull are thermodynamically stable. Configurations above the hull are metastable or unstable.

In ICET, after constructing a cluster expansion Hamiltonian and predicting energies for many configurations (via enumeration, random sampling, or MC simulations), the ConvexHull class can be used to compute and visualize the convex hull. The algorithm automatically identifies the set of

compositions and corresponding structures that form the ground-state phase diagram at zero Kelvin.

While this thesis focuses on finite-temperature thermodynamics (mixing free energy, and order-disorder phenomena), the convex hull provides an important zero-temperature reference. It serves several purposes:

- Validation of the cluster expansion: A physically reasonable convex hull confirms that the CE correctly captures the energetics of ordering tendencies.
- Benchmark for MC results: Stable phases predicted by the convex hull should match the trends seen in low-temperature Monte Carlo simulations.
- Insight into ordering: Ordered structures on the convex hull indicate possible ordered ground states at certain compositions, which may persist as ordered phases at higher temperatures or transform upon heating.

In the context of  $\text{In}_x\text{Ga}_{1-x}\text{N}$ , the convex hull allows us to identify whether there are any compositions where ordered ground states are predicted, or whether the system remains a disordered solid solution even at low temperatures.

### 1.5 Monte-Carlo Simulations

Monte Carlo (MC) simulations are a class of computational algorithms that rely on random sampling to obtain numerical results. They are particularly powerful in systems where the number of configurations is too large for exhaustive enumeration. In statistical mechanics, MC simulations are used to explore the high dimensional configurational space and compute thermodynamic averages over ensembles, providing a bridge between microscopic interactions and macroscopic observables. The central idea of MC methods is the generation of a representative set of configurations  $\{\sigma\}$  according to a probability distribution governed by the Boltzmann factor:

$$P(\sigma) = \frac{1}{Z} \exp\left(-\frac{E(\sigma)}{k_B T}\right) \quad (1.12)$$



where :

$$Z = \sum_{\sigma} \exp\left(-\frac{E(\sigma)}{k_B T}\right)$$

(1.13) is the partition function

The Metropolis algorithm, introduced in 1953, is one of the most widely used in MC methods. It works as follows:

1. Start from an initial configuration  $\sigma$
2. Choose randomly a new configuration  $\sigma'$  by a small modification (e.g. swapping atoms)
3. Calculate the energy difference  $\Delta E = E(\sigma') - E(\sigma)$  (1.14)
4. Accept the new configuration with probability :

$$P_{\text{accept}} = \min\left(1, \exp\left(\frac{-\Delta E}{k_B T}\right)\right) \quad (1.15)$$

5. If the move is rejected, keep the current configuration
6. Repeat the process to generate a Markov chain of configurations.

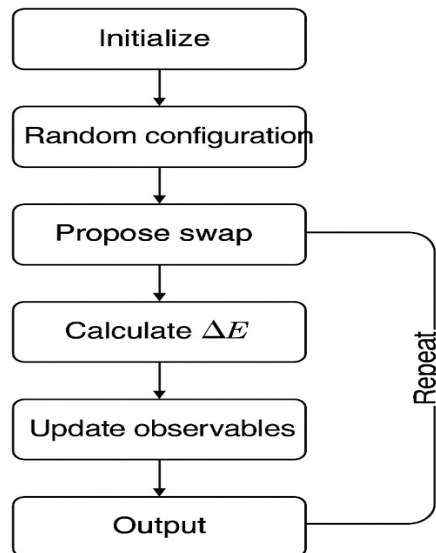


Figure 1.6 : Monte-Carlo simulation workflow using CE Hamiltonian

This process ensures detailed balance and convergence toward equilibrium distribution:

$$P(\sigma)W(\sigma \rightarrow \sigma') = P(\sigma')W(\sigma' \rightarrow \sigma) \quad (1.16)$$

where  $W(\sigma \rightarrow \sigma')$  is the transition probability.

### Thermodynamic Averages

Once equilibrium is reached, observables can be estimated as ensemble averages over sampled configurations

- **Internal Energy :**  $\langle E \rangle = \sum_{\sigma} P(\sigma) E(\sigma) \quad (1.17)$
- **Heat Capacity :**  $C_V = \frac{1}{K_B T^2} (\langle E^2 \rangle - \langle E \rangle^2) \quad (1.18)$
- **Order parameters:** Quantities that signal phase transitions or ordering tendencies.

In this thesis, Monte-Carlo using a CE-derived Hamiltonian describing the  $\text{In}_x\text{Ga}_{1-x}\text{N}$  alloy. The total energy is approximated by :

$$E(\sigma) = E_0 + \sum_{\alpha} J_{\alpha} \Phi_{\alpha}(\sigma) \quad (1.19)$$

The configurations are commonly sampled using MC techniques under various thermodynamic ensembles. The most common are :

- Canonical Ensemble : Fixed number of atoms, volume and temperature.
- Semi-Grand Canonical Ensemble: Composition and temperature can vary, while total number of atoms is fixed.

In the semi-grand canonical ensemble, the Hamiltonian is modified to include chemical potentials  $\mu$  :

$$H(\sigma) = E(\sigma) - \sum_i \mu_i N_i(\sigma) \quad (1.21)$$

where  $N_i$  is the number of atoms of species  $i$

Canonical Ensemble is used to our system. Thus, this allows simulation of phase stability across different compositions and temperature ranges. The calculated thermodynamic observables are then used to construct phase diagrams and identify order-disorder transitions, as shown in Figure 1.7 where we see a peak at the heat capacity value, indicating a phase transition at  $x=0.35$ .

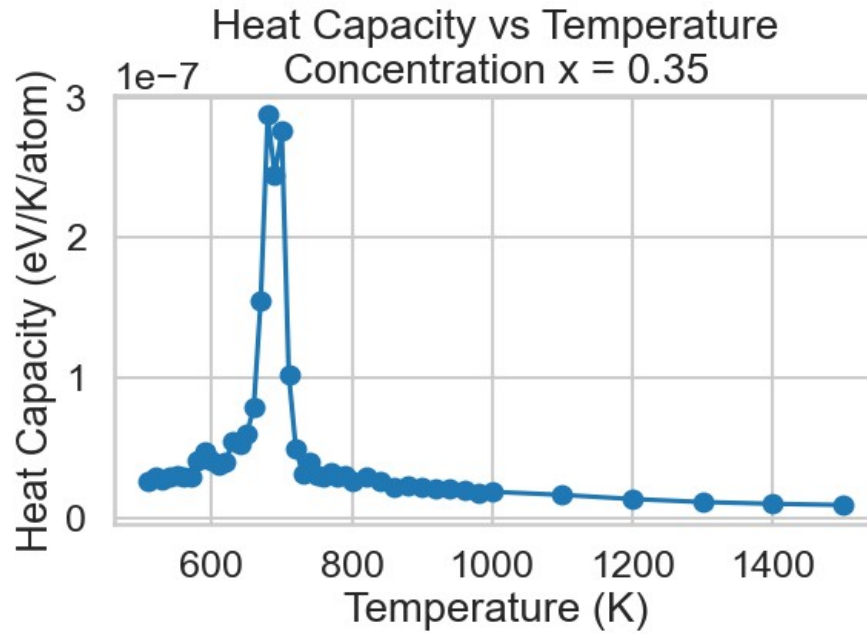


Figure 1.7 : Heat Capacity vs Temperature plot for a  $\text{In}_{0.35}\text{Ga}_{0.65}\text{N}$  alloy. We observe a peak in the heat capacity value in about  $T=700$  K, indicating phase transition in this configuration.

## 1.6 Warren-Cowley Short Order Range (SRO) Parameter

In alloys, atomic order often deviates from randomness due to energetic preferences between unlike and like pairs of atoms. Such deviations can be quantified by the Warren-Cowley short-range order (SRO) parameters, introduced by Warren and Cowley[citation]. The SRO parameter quantifies the tendency of unlike or like pairs of atoms to cluster or to repel one another in a given coordination shell.

### Definition

Let us consider a binary alloy  $\text{A}_x\text{B}_{1-x}$  with atomic fraction  $x$  of A atoms. Let  $\alpha_{ij}^{(n)}$  denote the SRO parameter between atom types  $i$  and  $j$  at the  $n$ -th neighbor shell (coordination shell). The parameter is defined as :

$$\alpha_{ij}^{(n)} = 1 - \frac{P_{ij}^{(n)}}{x_j} \quad (1.22)$$

where :

- $P_{ij}^{(n)}$  is the conditional probability of finding a j-type atom in the n-th neighbor shell of an i-type atom.
- $x_j$  is the overall concentration (atomic fraction) of species j

The SRO parameters are interpreted as follows :

- $\alpha_{ij}^{(n)} = 0$  : Random occupation of sites n
- $\alpha_{ij}^{(n)} < 0$  : Preference for unlike neighbors (ordering tendency).
- $\alpha_{ij}^{(n)} > 0$  : Preference for like neighbors (clustering tendency).

For example, a negative  $\alpha_{\text{In-Ga}}^{(1)}$  in the  $\text{In}_x\text{Ga}_{1-x}\text{N}$  alloy indicates a preference for In and Ga atoms to be nearest neighbors, suggesting a short-range ordering effect.

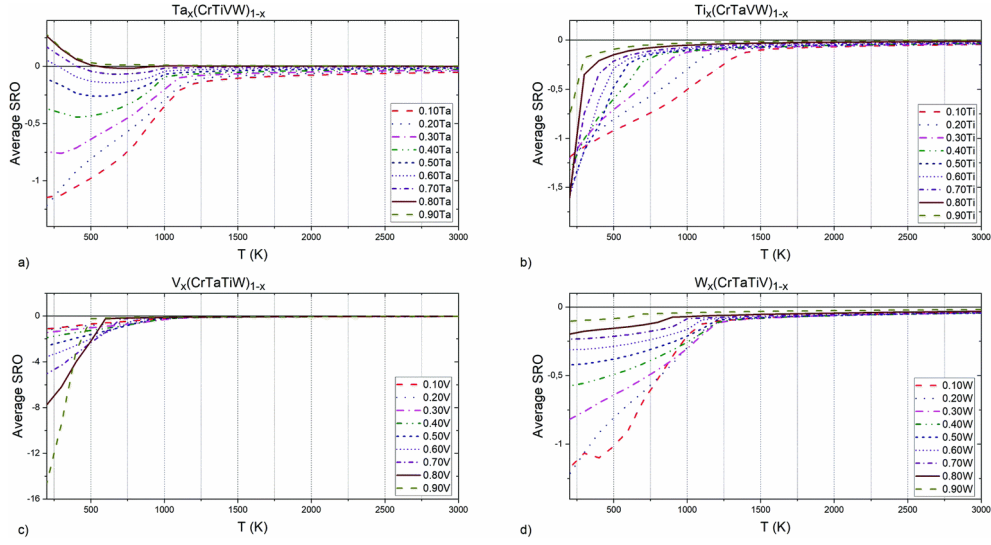
In practice we use the following form to calculate the SRO parameters:

$$\alpha_{ij}^{(n)} = 1 - \frac{N_{ij}^{(n)}}{Z^{(n)} x_j} \quad (1.23)$$

- $N_{ij}^{(n)}$  : Is the average number of j-type neighbors around an i-type atom in the n-th shell
- $Z^{(n)}$  : Denots the coordination number (number of neighbors in the n-th shell)

In simulations of alloy thermodynamics (e.g.,  $\text{In}_x\text{Ga}_{1-x}\text{N}$ ), the SRO parameter is often used to monitor the degree of chemical ordering at different temperatures and compositions. The parameter serves as an indicator of phase stability and local structure evolution, complementing global quantities such as mixing energy or entropy.

An example of the use of SRO parameters is shown in Fig (1.8) for various alloys.



**Figure 1.8 – Example of Warren–Cowley SRO Parameter Behavior in Refractory Alloys**

Chemical short-range order (SRO) parameters for various transition metal pairs in multi-component alloys, as a function of temperature. Subfigures (a–d) illustrate the influence of elemental concentrations on SRO trends for (a) Ta, (b) Ti, (c) V, and (d) W elements within Cr–Ti–V–W-based refractory systems. The Warren–Cowley  $\alpha$  values quantify the degree of local atomic ordering, with negative values indicating a preference for heteroatomic coordination and positive values suggesting like-atom clustering. These trends exemplify how alloy composition and temperature control the onset and evolution of chemical ordering — a principle similarly applied in this thesis to the  $\text{In}_x\text{Ga}_{1-x}\text{N}$  alloy system. [22]

The Warren–Cowley SRO parameters  $\alpha_{ij}^{(n)}$ , provide a clear, shell-resolved measure of how atomic arrangements deviate from randomness. In this thesis, SRO parameters will be extracted from canonical-ensemble MC simulations on the CE Hamiltonian, mapped as functions of temperature and composition to elucidate order–disorder phenomena in  $\text{In}_x\text{Ga}_{1-x}\text{N}$ .

### 1.7 Spinodal Decomposition

Spinodal decomposition is a mechanism of phase separation that occurs spontaneously when a homogeneous alloy enters a region of instability. This instability is characterized by a negative curvature of the free energy with respect to composition:

$$\left(\frac{\partial^2 F_{mix}}{\partial x^2}\right)_T < 0 \quad (1.31)$$

This defines the spinodal region, where even infinitesimal composition fluctuations reduce the free energy, causing the alloy to decompose into two phases of differing composition without any nucleation barrier.

The boundary of this region, where:

$$\left(\frac{\partial^2 F_{mix}}{\partial x^2}\right)_T = 0 \quad (1.32)$$

is the spinodal line. Outside the spinodal line, phase separation can still occur but requires overcoming an energy barrier via nucleation and growth.

#### Thermodynamic Pathways of Decomposition

- Inside the spinodal: phase separation occurs via continuous modulation of composition — typical of spinodal decomposition.
- Outside the spinodal but within the miscibility gap: decomposition proceeds via classical nucleation and growth.

At high temperatures, entropic stabilization can prevent phase separation. However, as temperature decreases, the balance between  $E_{mix}$  and  $TS_{mix}$  shifts, and demixing becomes thermodynamically favored in certain composition ranges.

These findings are consistent with prior ab initio-based thermodynamic investigations. For example, Duff et al. [1] used first-principles quasiharmonic calculations to show that strain significantly alters the Gibbs free energy landscape of  $\text{In}_x\text{Ga}_{1-x}\text{N}$ , suppressing phase separation and reducing the spinodal region. While their approach includes vibrational contributions and coherent

epitaxial effects, the present work focuses on configurational thermodynamics using Monte Carlo methods and cluster expansion, providing complementary insight into the alloy's intrinsic mixing behavior.

### 1.8 Phase Diagram Construction : Chemical Potentials and Spinodal Curves

A central goal of this work is to construct a comprehensive phase diagram for the  $\text{In}_x\text{Ga}_{1-x}\text{N}$  alloy, capturing the phase stability and thermodynamic behavior across a wide range of compositions and temperatures. The primary variable used for mapping the phase diagram is the chemical potential difference between indium and gallium atoms:

$$\Delta\mu = \mu_{\text{In}} - \mu_{\text{Ga}} \quad (1.33)$$

The phase diagram is represented in the  $\Delta\mu$ - $T$  plane. This section describes the methodology used to calculate  $\Delta\mu$  from Monte Carlo data and the extraction of binodal curves using a common tangent construction.

#### 1.8.1 Derivation of $\Delta\mu$ used in this work

The chemical potential  $\mu$  of species  $i$  is defined as :

$$\mu_i = \left( \frac{dG}{dN_i} \right)_{T, P, N_{i \neq j}} \quad (1.34)$$

Where  $G$  is the Gibbs free energy and  $N_i$ , the number of atoms  $i$ . The Gibbs free energy is defined as :

$$G = U + PV - TS \quad (1.35)$$

Where  $U$  is the internal energy,  $P$  and  $V$  the pressure and the volume and  $T$  and  $S$  the temperature and entropy. In our model we consider that the volume changes very small and the  $PV$  term can be neglected. The entropy  $S$  term, in principle includes configurational, electronic, magnetic and phononic. The Metropolis algorithm accounts for configurational entropy only. Therefore, it neglects the all-other contributions. However,  $\text{InGaN}$  are no-magnetic and are wide bandgap materials. Therefore, the corresponding contributions can also be neglected. **Regarding the phononic contributions these are typically small in solid state.** In our evaluation we are interested in energy differences. Although we treat  $\text{InGaN}$  as  $\text{InGa}$  pseudobinary alloys, the metal atoms are second nearest neighbors in the wurtzitic crystal. Metal-metal interactions are expected to have

even smaller effect on differences in the phononic contributions to entropy. Therefore, these contributions are also neglected in our evaluation.

Our MC calculations provide mixing free energies as function of temperature and composition

The starting point is the expression for the Gibbs free energy per atom of the alloy:

$$G(x, T) = E_{mix}(x, T) + xU_{InN}(T) + (1-x)U_{GaN}(T) \quad (1.36)$$

The internal energies  $U_{InN}(T)$  and  $U_{GaN}(T)$  depend on  $T$ . However we neglect entropic contributions other than configurational. Therefore, their dependence on  $T$  can be dropped.

The chemical potential difference is formally defined as :

$$\Delta\mu = \mu_{In} - \mu_{Ga} = 2 \frac{dG}{dx} \quad (1.37)$$

The factor of 2 arises because an infinitesimal change in composition  $dx$  corresponds to exchanging one In atom for one Ga atom in the binary alloy. This exchange process corresponds to a net chemical potential difference involving both species, hence the factor of 2.

Taking the derivative of  $G(x, T)$  with respect to  $x$  gives:

$$\frac{dG}{dx} = \frac{dE_{mix}}{dx} + U_{InN} - U_{GaN} \quad (1.38)$$

Thus, the chemical potential difference becomes:

$$\Delta\mu = 2 \left( \frac{dE_{mix}}{dx} + U_{InN} - U_{GaN} \right) \quad (1.38)$$

In practice, the derivative  $\frac{dE_{mix}}{dx}$  was computed numerically using a finite difference:

$$\frac{dE_{mix}}{dx} \approx \frac{E_{mix}(x) - E_{mix}(x - \Delta x)}{\Delta x} \quad (1.39)$$

with  $\Delta x = 0.05$ . The resulting expression implemented in the analysis script was therefore :

$$\Delta\mu = 2 \left( \frac{E_{mix}(x) - E_{mix}(x - \Delta x)}{\Delta x} \right) + 2\Delta U \quad (1.40)$$



This approach allows direct computation of  $\Delta\mu(x,T)$  from the MC simulation results. The resulting  $\Delta\mu$  values were interpolated and visualized as a  $\Delta\mu$ – $T$  phase diagram, forming the central thermodynamic result of this thesis.

To identify phase boundaries and determine the miscibility gap of the  $\text{In}_x\text{Ga}_{1-x}\text{N}$  alloy, spinodal curves were extracted using a common tangent construction applied to the free energy of mixing  $E_{\text{mix}}(x,T)$

The thermodynamic condition for phase coexistence between compositions  $x_1(T)$  and  $x_2(T)$  is that their free energies satisfy the common tangent condition:

$$\left. \frac{dE_{\text{mix}}}{dx} \right|_{x_1} = \left. \frac{dE_{\text{mix}}}{dx} \right|_{x_2} \quad (1.42)$$

and

$$E_{\text{mix}}(x_2) - E_{\text{mix}}(x_1) = \left( \frac{dE_{\text{mix}}}{dx} \right) (x_2 - x_1) \quad (1.43)$$

In practice, this condition identifies pairs of compositions that can coexist in thermodynamic equilibrium at the given temperature.

In this work, due to the rather sparse mesh used to sample different compositions, we estimate the points where the second derivative of the mixing energy with respect to  $x$  is negative.

## Chapter 2

### Results and Discussion

This chapter presents the computational results obtained for the  $\text{In}_x\text{Ga}_{1-x}\text{N}$  alloy system using the cluster expansion and Monte Carlo approaches described in Chapter 1. The primary goal of this study is to analyze the configurational and thermodynamic behavior of the alloy across a wide range of compositions and temperatures.

We first evaluate the quality of the constructed cluster expansion and examine the zero-temperature stability of the alloy via the convex hull of the formation mixing energies. Subsequently, finite-temperature properties are explored through extensive MC simulations. These simulations provide key thermodynamic quantities, including the mixing free energy, which are used to identify phase stability trends. Additionally, short-range order parameters are analyzed to reveal atomic ordering tendencies within the alloy.

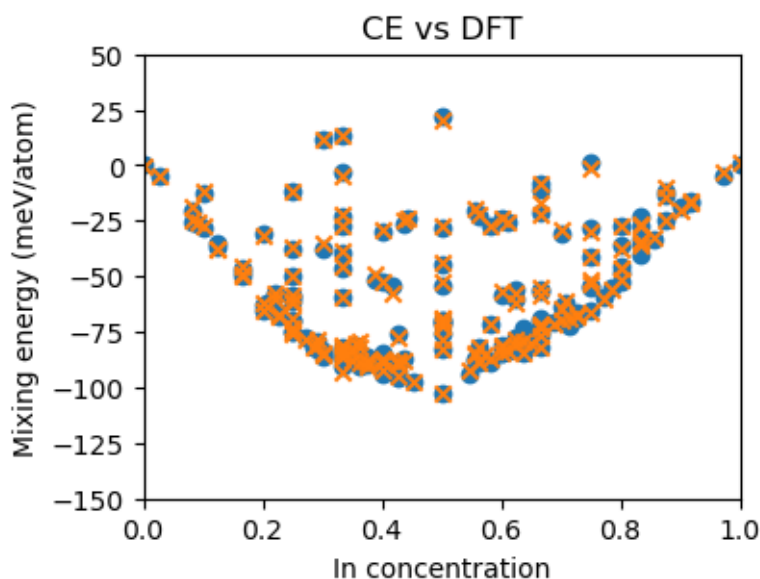
The results presented in this chapter aim to provide a comprehensive understanding of the thermodynamic landscape of  $\text{In}_x\text{Ga}_{1-x}\text{N}$ , offering insights into phase stability, mixing behavior, and the role of ordering phenomena under varying thermodynamic conditions.

#### 2.1 Cluster Expansion Results

The first step in the thermodynamic analysis of the  $\text{In}_x\text{Ga}_{1-x}\text{N}$  alloy system was the construction of an accurate cluster expansion (CE) Hamiltonian, which serves as the core of the Monte Carlo simulations performed in this study. The CE was trained using formation energies obtained in a previous work from density functional theory (DFT) calculations on a set of 160 atomic configurations spanning the full composition range of the alloy. The calculations performed within the Local Density Approximation (LDA) using VASP [citation]. Pseudomorphic growth conditions considered for these calculations, i.e., InGaN alloys are biaxially strained to GaN. This section presents the quality of the resulting CE model and its predictions for zero-temperature phase stability.

### CE Fitting Quality

The accuracy of the constructed cluster expansion (CE) was evaluated by directly comparing the mixing energies predicted by the CE to those obtained from first-principles (DFT) calculations across all range of In concentrations. Figure 2.1 shows the mixing energy of each configuration as a function of In concentration, with DFT results and CE predictions plotted simultaneously for comparison.



"Figure 2.1: Comparison of DFT and CE calculated mixing energies as a function of In concentration in  $\text{In}_x\text{Ga}_{1-x}\text{N}$  alloys."

The two sets of data are in excellent agreement throughout the entire composition range. The CE model successfully captures both the overall trend and the fine configurational variations of the mixing energies curve. No significant systematic deviations are observed, indicating that the CE accurately reproduces the underlying configurational energetics of the alloy.

Quantitatively, the root-mean-square error (RMSE) of the CE with respect to the DFT reference energies is approximately 1.83 meV/atom. This level of accuracy is fully sufficient for subsequent MC simulations and thermodynamic analysis.

To further evaluate the quality of the cluster expansion across the entire composition range, the residuals — defined as the difference between DFT-calculated and CE-predicted mixing energies — were plotted as a function of In concentration (Fig. 2.2).

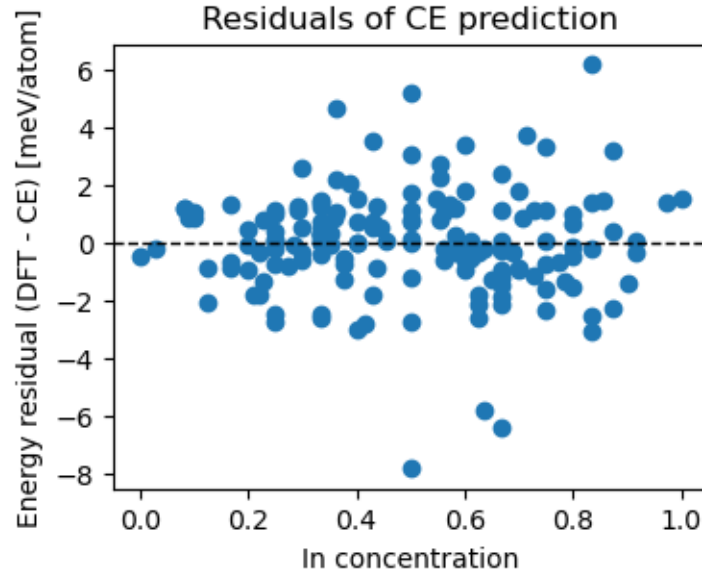


Figure 2.2: Residuals of CE predicted mixing energies with respect to DFT reference energies, plotted as a function of In concentration in the  $\text{In}_x\text{Ga}_{1-x}\text{N}$ .

The residuals are uniformly small, with no significant systematic bias or trend. This confirms that the CE accurately captures both the global and local variations of the configurational energetics in the  $\text{In}_x\text{Ga}_{1-x}\text{N}$  alloy. The magnitude of the residuals remains well below 10 meV/atom for nearly all configurations, ensuring that the CE is sufficiently accurate for subsequent thermodynamic analysis and Monte Carlo simulations. Comparing this value to  $k_B T$ , we estimate that this correspond to less than 100 K error in templates. It is important to note that for the technologically relevant compositions for optoelectronic devices in the green area of the spectrum the residuals are below 4 meV/atom.

### Convex Hull of Formation Energies

In order to assess the zero-temperature phase stability of the  $\text{In}_x\text{Ga}_{1-x}\text{N}$  alloys and to further validate the accuracy of the CE, the convex hull of formation energies was computed using the trained CE Hamiltonian. The formation energies of a large set of configurations were predicted and the convex hull was constructed using the ICET ConvexHull class

Fig. 2.3 shows the resulting convex hull. The scatter of blue crosses represents the predicted mixing energies of 333256 configurations considered, while the green line denote the convex hull. Configurations lying on the convex hull are therefore thermodynamically stable at  $T = 0$  K.

Configurations lying above the hull are metastable and would decompose into a mixture of phases lying on the hull.

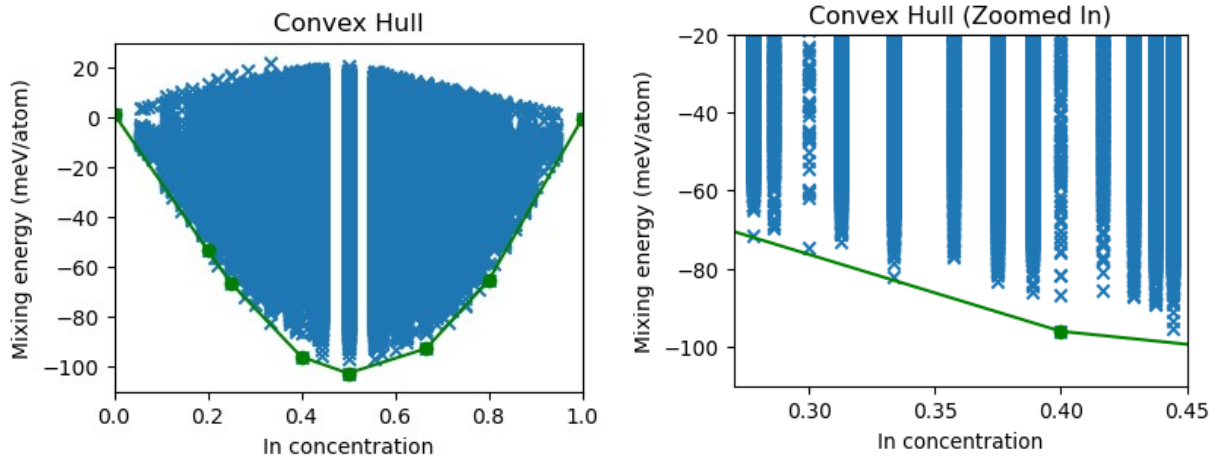


Figure 2.3: Convex hull of formation energies computed from the cluster expansion for the  $\text{In}_x\text{Ga}_{1-x}\text{N}$  alloy. Blue crosses represent the predicted mixing energies of all configurations; the green line denote convex hull. Right: Zoom into the region of technologically important In contents for optoelectronics in the green spectral region.

The mixing energies and the convex hull shown in Fig(2.3) have a single minimum at  $x=0.5$ . This indicates that at  $T=0$  the alloys is stable against decomposition towards the its end constitutens, i.e., GaN and InN. This is consistent with experimental observations and previous theoretical studies, which show that epitaxial strain tends to suppress phase separation in the InGa<sub>N</sub> system.

**Another important outcome of this diagram is related to the technologically relevant compositional region between  $x=0.3$  and  $x=0.4$  (see Fig(2.3) right). In this region we found ground states at  $x=0.35$  and  $0.4$ . For compositions between these values all structures have mixing energies well above the convex hull. This indicates a region where InGa<sub>N</sub> alloys are prone to decompose towards structures with  $x=0.35$  and  $0.40$ .**

## 2.2 Monte Carlo Calculations

Monte Carlo (MC) simulations were carried out using the MCHammer module from the ICET package to investigate thermodynamic and structural behavior across temperature for a range of compositions in the  $\text{In}_x\text{Ga}_{1-x}\text{N}$  alloy. All simulations were performed in the canonical ensemble

using a cluster expansion Hamiltonian trained on density functional theory (DFT) calculations. The following parameters summarize the simulation setup:

All simulations were performed on a 1000-atom supercell constructed by repeating the wurtzite primitive unit cell along the  $x$ ,  $y$ , and  $z$  directions in a  $10 \times 10 \times 5$  configuration. For a given composition  $x$ , a fraction  $x$  of the Ga atoms was randomly substituted with In atoms on the cation sublattice, while N atoms were left unchanged.

### Composition Mesh

To resolve composition-dependent effects, simulations were performed over a composition mesh ranging from  $x=0.05$  to  $x=0.95$  in increments of 0.05, yielding a total of 19 unique compositions. Each composition was treated independently, with a random initial distribution of In atoms on the cation sites consistent with the target concentration.

### Temperature Grid and Annealing

Temperature-dependent simulations were carried out for each composition over a finely resolved temperature grid:

- From 1800 K to 800 K in steps of 20 K
- From 800 K to 0 K in steps of 5 K

This two-stage temperature schedule allows for efficient sampling of high-temperature behavior while providing fine resolution in the low-temperature regime where ordering phenomena and phase transitions may occur.

Simulations proceeded via annealing, where the final configuration at a given temperature was used as the initial configuration for the next lower temperature. This gradual cooling strategy accelerates equilibration and improves consistency across temperature steps.

### Monte Carlo Sampling

At each temperature, the system underwent a total of  $N^2$  trial swaps, where  $N$  is the number of atoms in the supercell. For a 1000-atom system, this corresponds to 1,000,000 Monte Carlo steps per temperature. To ensure statistical reliability, only the last half of the trajectory (i.e., the final 500,000 steps) was used for ensemble averaging, while the first half was discarded as equilibration.

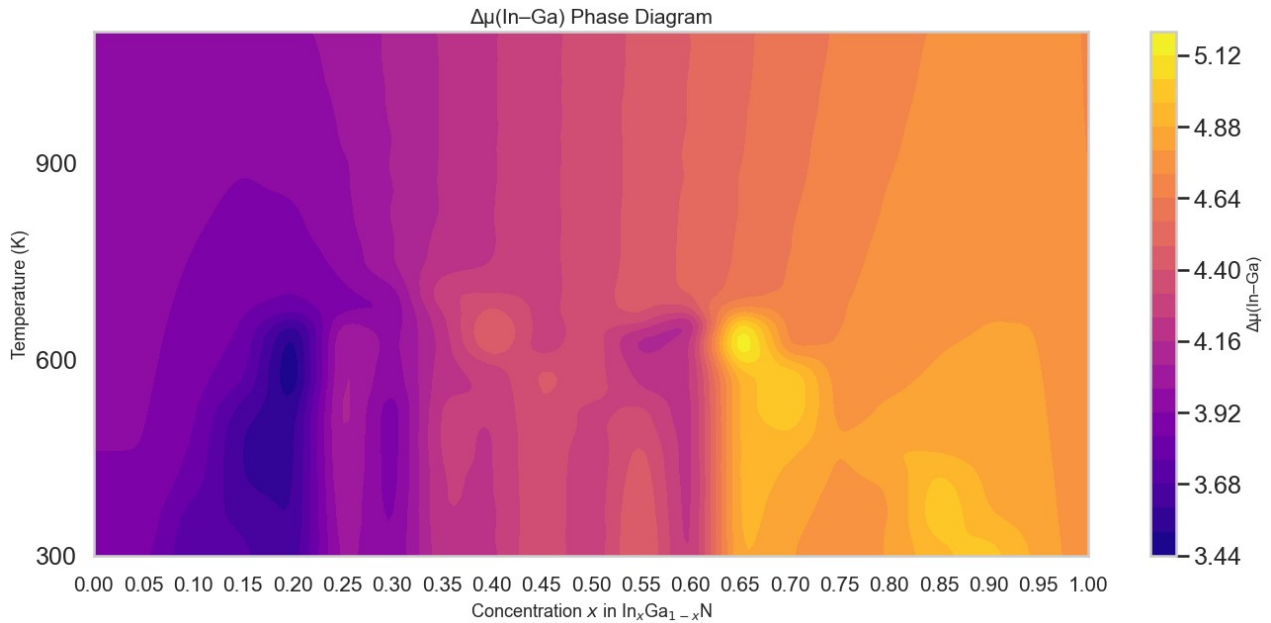
This choice of step count—scaling quadratically with system size—was selected to ensure sufficient sampling of configuration space, especially in low-temperature regimes where equilibration slows and local energy minima dominate the landscape.

### 2.3 Phase Diagram of $\text{In}_x\text{Ga}_{1-x}\text{N}$ Alloy

The central result of this study is the construction of a detailed phase diagram for the  $\text{In}_x\text{Ga}_{1-x}\text{N}$  alloy, expressed in terms of the chemical potential difference  $\Delta\mu = \mu_{\text{In}} - \mu_{\text{Ga}}$  plotted against composition  $x$  and  $T$ . This representation provides a direct thermodynamic map of phase stability and ordering tendencies in the alloy system.

#### $\Delta\mu$ - $T$ Phase Diagram

The starting point of the phase diagram analysis is the chemical potential difference  $\Delta\mu = \mu_{\text{In}} - \mu_{\text{Ga}}$ , computed directly from the Monte Carlo simulation data as described in Section 1.8.1. The resulting  $\Delta\mu(x,T)$  values provide a direct thermodynamic map of the alloy system across temperature and composition. The raw  $\Delta\mu(x,T)$  data were interpolated onto a dense grid in  $(x,T)$  space and smoothed to produce a smooth diagram. The resulting map is shown in Figure 2.2.1



**Figure 2.2.1:** Phase diagram of the  $\text{In}_x\text{Ga}_{1-x}\text{N}$  alloy. The color map represents the chemical potential difference  $\Delta\mu(\text{In-Ga})$  plotted as a function of composition and temperature.

### Phase Transition Points from Heat Capacity Peaks

To identify phase transition temperatures in the  $\text{In}_x\text{Ga}_{1-x}\text{N}$  alloy, the heat capacity  $C_V$  was computed from MC simulations using energy fluctuations at each temperature and composition:

$$C_V(T) = \frac{\langle E^2 \rangle - \langle E \rangle^2}{k_B T^2}$$

The temperature dependence of  $C_V(T)$  provides a sensitive probe of phase transitions. Sharp peaks in  $C_V(T)$  typically correspond to order-disorder transitions or the onset of phase separation.

Fig. (2.2.3) shows the  $C_V(T)$  curves for all alloy compositions that the heat capacity shows a discontinuity. The temperatures corresponding to the peaks in  $C_V(T)$  were extracted and correspond to order-disorder phase transition points.

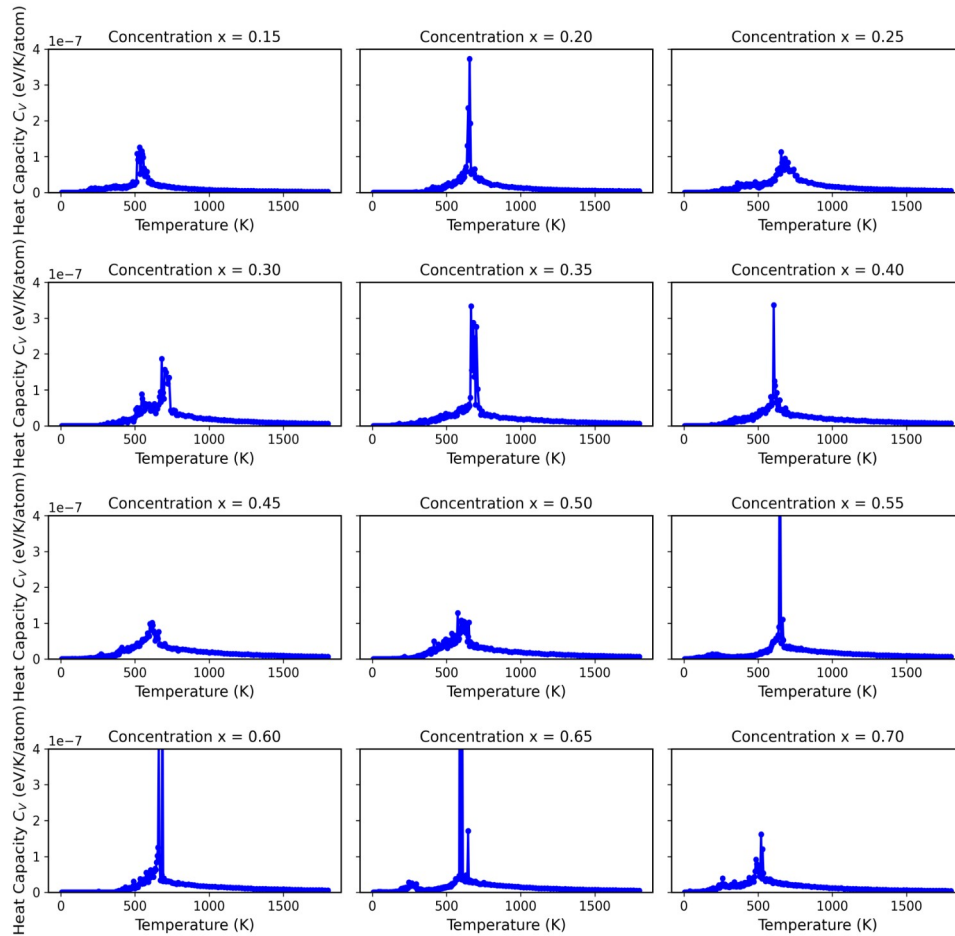




Figure 2.2.3: Heat capacity  $C_V(T)$  vs  $T$  curves for all compositions of  $\text{In}_x\text{Ga}_{1-x}\text{N}$  where the heat capacity shows a peak. The peaks indicate critical temperatures associated with order-disorder transitions.

The resulting phase transition points were then superposed on the  $\Delta\mu$ - $T$  phase diagram to visualize order-disorder boundaries (Figure 2.2.4).

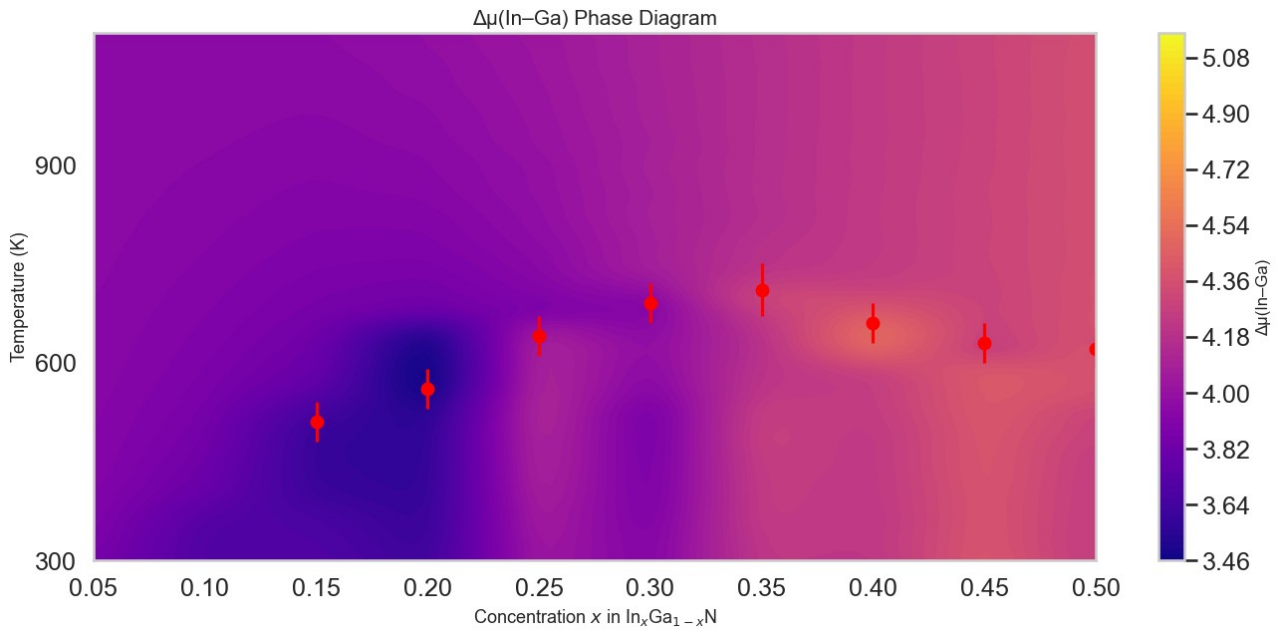


Figure 2.2.4: Phase transition points (red markers) identified from the maxima in heat capacity  $C_V(T)$  across the composition range, superposed on the chemical potential difference ( $\Delta\mu(\text{In-Ga})$ ) phase diagram. The contour plot represents the variation of  $\Delta\mu(\text{In-Ga})$  with composition ( $x$ ) and temperature, highlighting and critical transition regions within the  $\text{In}_x\text{Ga}_{1-x}\text{N}$  alloy system.

The  $C_V(T)$  peak temperatures closely track the temperatures at which significant changes in the alloy's configurational state occur, making them a valuable guide for mapping ordered phase stability.

In addition to the phase transition points identified from heat capacity peaks, regions where the alloys is unstable against decomposition were identified

In this study, however, the spinodal points were not obtained via automated second derivative calculations. Therefore, a visual approach was employed. For each temperature, the  $E_{\text{mix}}(x,T)$  curve was plotted, and pairs of compositions exhibiting an approximately equal slope — corresponding to a common tangent — were identified by inspection. Due to limited computational resources, we are restricted to a rather small set of compositions (0., 5%, 10%, ... etc.). these are sufficient to produce the phase diagram and highlight regions of spinodal decompositions. These are regions of small density of controur lines or no contour lines. However, such a sparse mesh of compositions is not adequate to numerically calculate second derivatives of the mixing free energy.

Although approximate, and indeed it can provide only a qualitative description, this visual common tangent method captures the essential physics. The resulting  $\Delta\mu$ -T phase diagram, is shown in Figure 2.2.5.

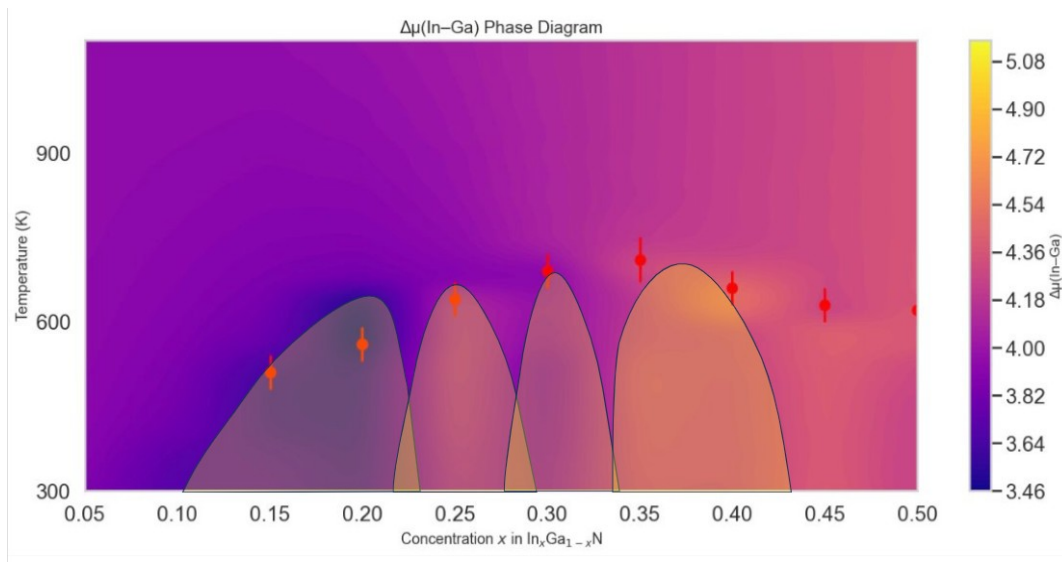


Figure 2.2.5: As in Fig. 2.2.4. The shaded areas denote approximately regions prone to decomposition. These regions were plotted based on the points of negative second order derivative of the mixing energy with respect to to composition and the lack of contour lines.

The phase diagram in Fig 2.2.5, reveals that. At high temperatures,  $\Delta\mu$  varies smoothly with temperature, driven primarily by entropy effects. However temperature decreases, order-disorder transitions become evidences from the discontinuities in the heat capacity. at the region of 30%-40% In content these appear at approximately 700 K. At lower In contents phase transitions are shifted to lower T. For x less than 15% no order-disorder transition are present.

Furthermore, at this region of compositions areas with low density or no contour lines exist. These are also constrained and denoted by the estimated points with negative second derivative in the mixing free energy with respect to x. These are areas of spinodal decomposition and are denoted with the shaded region in the phase diagrams. In these areas and if kinetic barriers are low enough to allow efficient diffusion of In and Ga atoms, the system is unstable against decomposition towards composition at their left and right limits.

Therefore, InGaN alloys pseudomorphically grown on GaN, are not unstable against decomposition towards GaN and InN. However, certain composition constrained by the aforementioned boundaries, are unstable against decomposition towards intermediate In contents. Nevertheless, it should be noted here, that this analysis assumes that thermodynamic equilibrium could be established. This required, either high T and/or low kinetic barriers for In and Ga atoms that would allow for efficient mobility of In and Ga atoms. According to the best of our knowledge, diffusion constants for In and Ga atoms in InGaN alloys have not been reported and their calculations goes far beyond the scope of a degree thesis.

## 2.4 Configurational Properties

The phase diagram discussed in the previous Section indicates regions where ordered phases of InGaN alloys are stable. Of particular interest is the nature of the ordering that develops below the critical temperature, where configurational entropy is reduced.

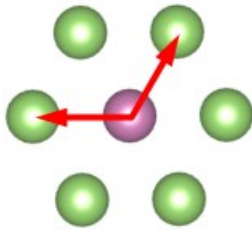
Understanding the nature of this atomic ordering is essential for investigating the electronic properties of these alloys. In this section, we investigate the ordering behavior of  $\text{In}_x\text{Ga}_{1-x}\text{N}$  within the technologically relevant compositional range for optoelectronic devices in the optical region of the spectrum ( $x = 0$  to  $0.40$ ) using both visual inspection of atomic configurations derived from MC calculations and quantitative short-range order (SRO) analysis. This combined approach allows us

to identify specific motifs, determine their dependence on composition and temperature, and connect microscopic ordering trends to the macroscopic features of the phase diagram.

### Definition of Neighbor Shells

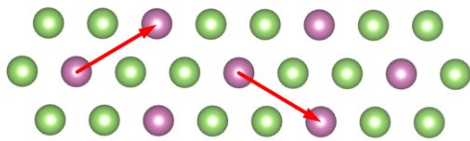
To analyze atomic ordering in  $\text{In}_x\text{Ga}_{1-x}\text{N}$  alloys, we classify neighbor interactions according to distance and crystallographic direction. In this study, we focus on four specific types of atomic neighbors: first and second nearest neighbors in the basal plane, i.e., the (0001) plane and first and second nearest neighbors located at different layers along [0001]. These are schematically shown in the Figs. below

**Figure 2.4.1 :**



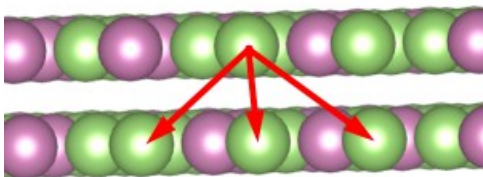
*First nearest neighbors* in the (0001) planes. Each atom has six such neighbors in a hexagonal planar configuration. We denote the corresponding SRO parameters as  $\alpha_a$

**Figure 2.4.2 :**



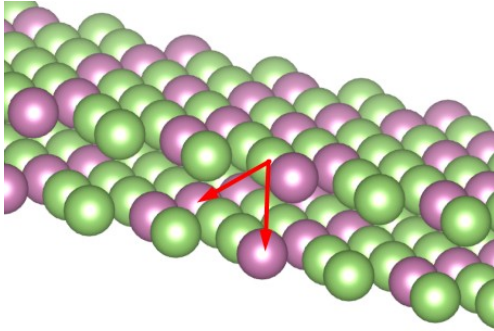
*Second nearest neighbors* in the same basal plane. We denote the corresponding SRO parameters as  $\alpha_m$

**Figure 2.4.3 :**



*First nearest neighbors* that lie in adjacent atomic layers above or below the reference atom, along the [0001] direction. we denote the corresponding SRO parameters as  $\alpha_{m+c}$

**Figure 2.4.4 :**



*Second nearest neighbors* that lie in next-nearest (0001) planes. We denote the corresponding SRO parameters as  $\alpha_c$

### Tendency to Form Specific Ordering

In the bulk  $\text{In}_x\text{Ga}_{1-x}\text{N}$  alloy system, atomic ordering arises not from direct chemical interactions between In and Ga atoms—since both species bond exclusively to nitrogen—but rather from strain-mediated interactions induced by the significant size mismatch between In–N and Ga–N bonds. These elastic interactions are known to can generate an effective repulsion between In atoms embedded in the GaN matrix.

As described in *Lee et al.* (Phys. Rev. B 90, 245301), these strain effects in surface layers result in ordering motifs where In atoms are preferentially aligned as second nearest neighbors in the basal plane. This result in a  $\sqrt{3}\times\sqrt{3}$  symmetry (see Fig. (2.4.5)). Nevertheless, these calculations where based on empirical potentials and described the surfaces of these alloys only. In the present thesis we are based on DFT derived CE Hamiltonian and we investigate the bulk thermodynamics of these alloys.

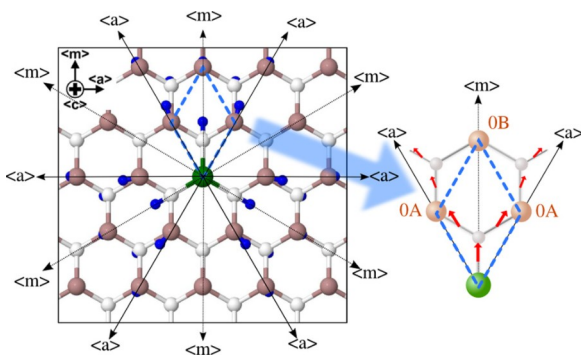


Figure 2.4.5 — In-plane (c-plane) displacement of Ga and N atoms induced by the incorporation of an In atom into bulk GaN. The  $\langle a \rangle$  crystallographic direction, highlighted in the figure, marks the primary axis along which strain relaxation occurs via c channeling of forces through In–N and Ga–N bonds (red arrows). These displacements generate effective repulsion between In atoms aligned along  $\langle a \rangle$ , while

positions along  $\langle m \rangle$  are strain-relieving and

energetically favorable for further In incorporation. This mechanism explains the emergence of ordering motifs, observed in the bulk  $\text{In}_x\text{Ga}_{1-x}\text{N}$  supercells studied here. [23]

To investigate how the strain-driven ordering motifs evolve with temperature, we analyze the Warren–Cowley short-range order (SRO) parameters for first- and second-nearest neighbor shells, both in and out the basal plane, across compositions ranging from  $x = 0.20$  to  $x = 0.40$ . For each composition, we identify the order–disorder transition temperature ( $T_c$ ) by locating discontinuities or inflection points in the SRO vs. temperature plots. Notably, these discontinuities consistently align with the critical temperatures previously determined from heat capacity peaks (as discussed in Section 2.3)

Below the critical temperature  $T_c$ , the alloy exhibits ordering features: negative SRO values, reflect a strong tendency to avoid first-neighbor In–In contacts and instead stabilize motifs such as  $\sqrt{3} \times \sqrt{3}$  in-plane arrangements. Above  $T_c$ , configurational entropic contributions overcomes the energy benefit of ordering, and SRO values approach zero, signaling the loss of spatial correlations and the transition into a disordered phase. This combined analysis of temperature-resolved SRO and structural snapshots enables a detailed characterization of the ordering–disordering process and its connection to the alloy’s thermodynamic behavior.

In the wurtzite crystal structure, the  $\langle a \rangle$  and  $\langle m \rangle$  directions lie within the basal (0001) plane, corresponding to the  $[1\bar{1}-20]$  and  $[\bar{1}1-00]$  directions, respectively. The  $\langle c \rangle$  direction denotes the vertical  $[0001]$  axis, along which atomic layers are stacked.

### **$x=0.20$**

At composition  $x = 0.20$ , the Warren–Cowley SRO parameters exhibit a clear transition within the temperature range of **550–650 K**. This behavior marks the onset of disorder and coincides with the critical temperature derived from the heat capacity peak, which falls within this same range and it is  $T=560$  K (see Fig 2.2.4)

The three SRO parameters plotted against  $T$  in Fig. 2.4.7, take the value of  $-0.25$  at low  $T$  and approach the value of  $0$  as  $T$  increases. To visualize the corresponding structures the atomic configurations are shown in Fig. 2.4.6. As can be seen in Figs 2.4.6 (a),(b) at  $T=300$ K the In atoms arrange as second nearest neighbors both in the basal plane as well as along  $c$ -axis. Nevertheless, at  $T=740$  K, i.e., above the critical temperature, ordering cannot be observed (see Fig. 2.4.6(d)) in agreement with the negligible values of the SRO parameters at that temperature. At intermediate temperatures, ( $T= 570$  K , just above  $T_c$ , see Fig. 2.4.6(c)), the configuration shows partial degradation of the previously ordered state. Nevertheless, preferential distribution of In atoms as second nearest neighbors along  $[1-100]$  can still be seen.

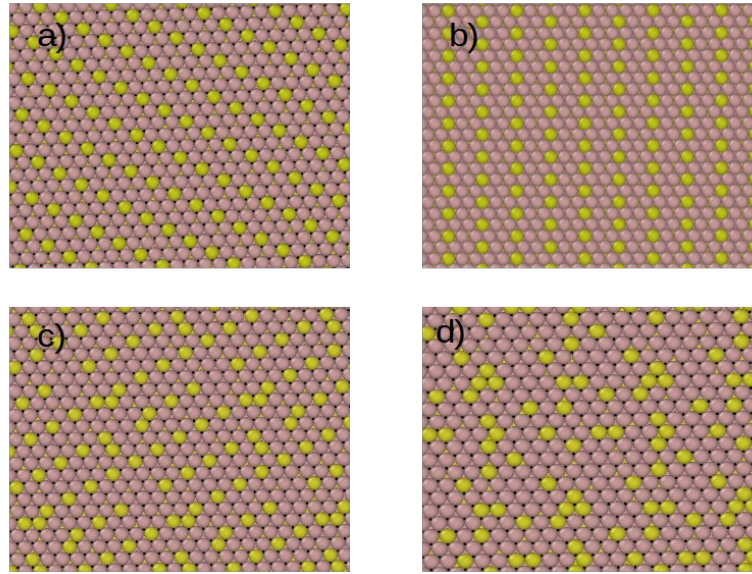


Figure 2.4.6 :Atomic configurations of  $\text{In}_{0.20}\text{Ga}_{0.80}\text{N}$  : (a) and (b) 300 K, (c) 570 K (just above  $T_c$ ), and (d) 740 K (well above  $T_c$ ). (a), (c), and (d) is a top view of the basal plane, and (b) is a side view. Yellow balls are In atoms and brownish balls are Ga atoms.

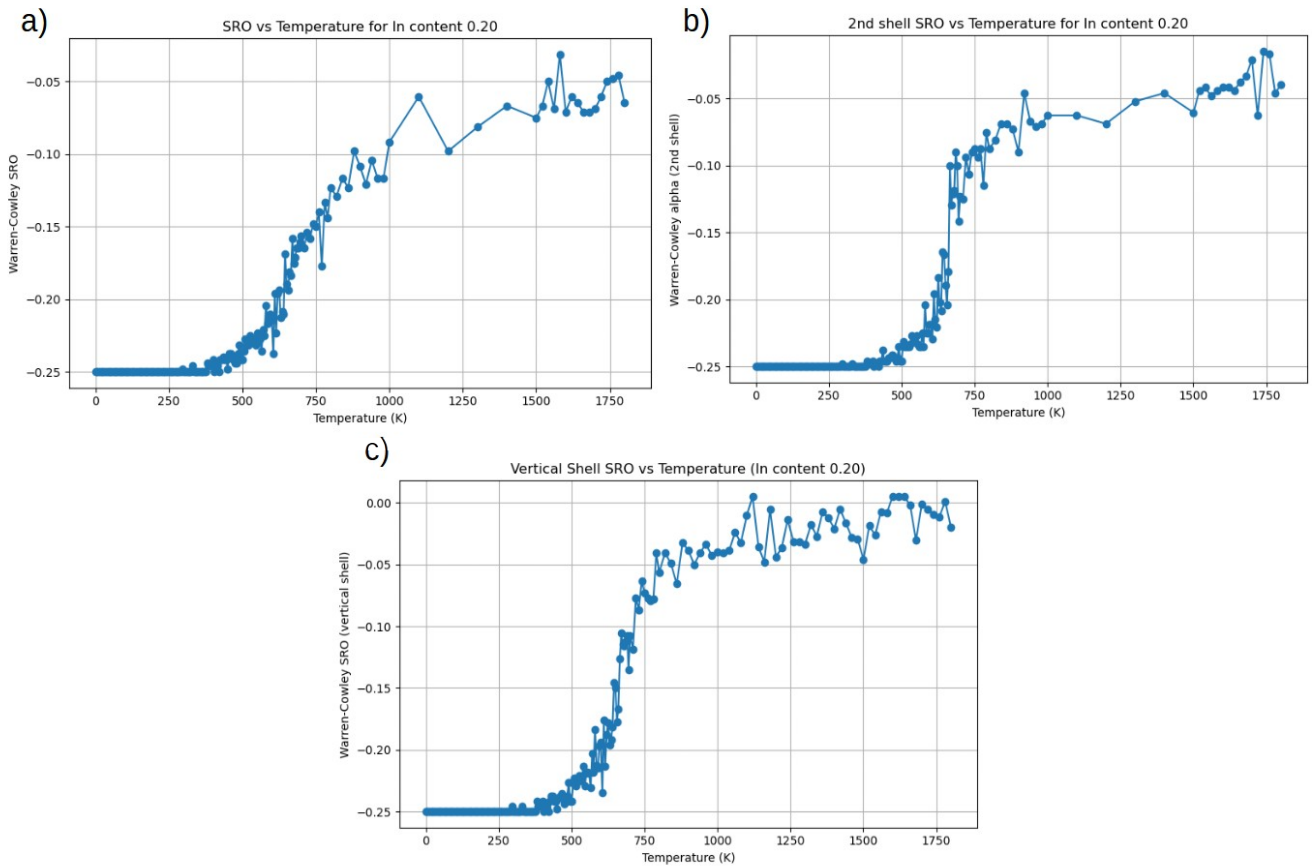




Figure 2.4.7 : Warren–Cowley short-range order (SRO) parameters as a function of temperature for  $\text{In}_{0.20}\text{Ga}_{0.80}\text{N}$ . (a)  $\alpha_a$ , (b)  $\alpha_m$ , (c)  $\alpha_c$

**x=0.25**

In this composition, the Warren–Cowley SRO parameters reveal a pronounced transition within the temperature range of 560–660 K, signaling the onset of disorder. This transition coincides with the critical temperature  $T_c$  derived from the heat capacity peak, which lies within the same range ( $T_c \approx 640$  K, see Fig 2.2.4)

The three SRO parameters plotted in Fig. 2.4.9 follow the expected behavior: they begin near  $-0.30$  at low temperatures and progressively approach zero as the temperature increases. The corresponding atomic configurations are shown in Fig. 2.4.8. At  $T = 300$  K, the In atoms exhibit clear second-neighbor alignment in a  $\sqrt{3} \times \sqrt{3}$  pattern both in the basal plane and along the vertical direction, as observed in Figs. 2.4.8(a), (b). At  $T = 840$  K, well above the critical point, ordering is no longer apparent [Fig. 2.4.8(d)], consistent with the near-zero SRO values at that temperature. At an intermediate state,  $T = 650$  K, just above  $T_c$  [Fig. 2.4.8(c)], the atomic configuration shows partial degradation of the ordered motifs. Still, residual second-neighbor In–In correlations, especially along  $[110]$ , remain noticeable, indicating a gradual loss of order rather than an abrupt disappearance.

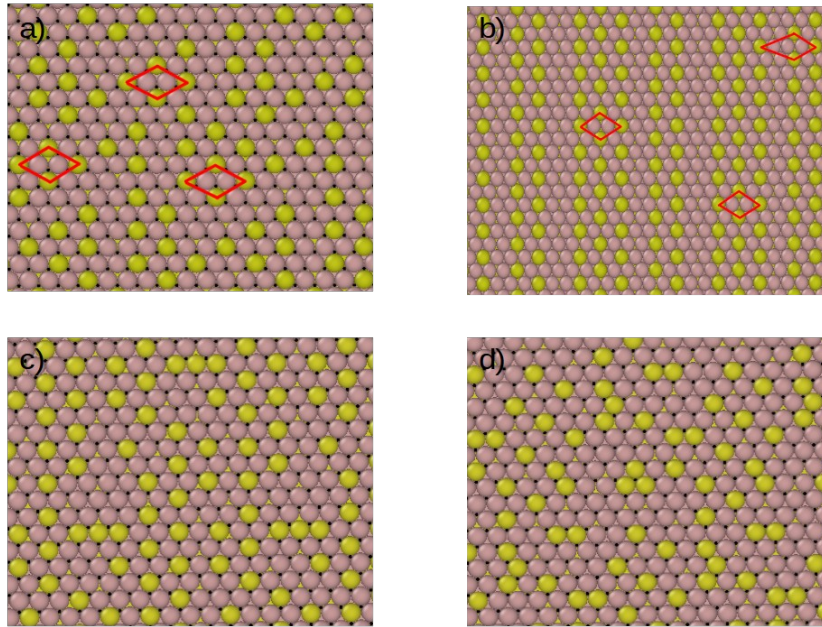


Figure 2.4.8: Atomic configurations of  $\text{In}_{0.25}\text{Ga}_{0.75}\text{N}$  at different temperatures: (a) and (b) 300 K, (c) 590 K (just above  $T_c$ ), and (d) 840 K (well above  $T_c$ ). (a), (c), and (d) is a top view of the basal plane, and (b) is a side view. Yellow balls are In atoms and brownish balls are Ga atoms.

At 300 K, In atoms form ordered arrangements characterized by second-nearest neighbor alignment both in-plane and along the  $c$ -axis,  $\sqrt{3} \times \sqrt{3}$  motifs outlined in red.



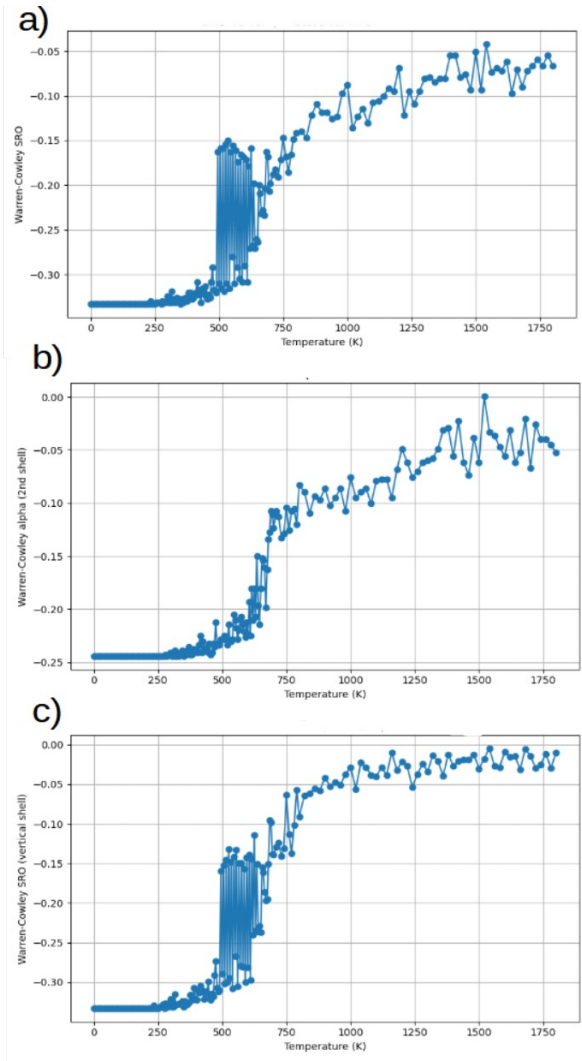


Figure 2.4.9:  
Warren–Cowley short-range order (SRO)  
parameters as a function of temperature for  
 $\text{In}_{0.25}\text{Ga}_{0.75}\text{N}$ . (a)  $\alpha_a$ , (b)  $\alpha_m$ , (c)  $\alpha_c$

### $x=0.30$

At composition  $x = 0.30$ , the Warren–Cowley SRO parameters demonstrate a pronounced transition between approximately 650–750 K, marking the breakdown of ordered correlations. This transition range aligns with the critical temperature of 690 K determined from the corresponding heat capacity peak (see Fig. 2.2.4), affirming the thermodynamic basis of the observed order–disorder transformation. As shown in Fig. 2.4.11, the three SRO curves begin at a strongly negative value near  $-0.40$  and progressively rise toward zero with increasing temperature, reflecting the gradual loss of local ordering. Atomic configurations at different temperatures are illustrated in Fig. 2.4.10. At  $T = 300$  K, clear second-neighbor In–In alignments are observed both in the basal plane and along the  $c$ -axis (Figs. 2.4.10(a,b)), consistent with the low-temperature SRO values. Just above the transition temperature, at  $T = 700$  K, the configurations (Fig. 2.4.10(c)) show visible degradation of these motifs, though remnants of second-neighbor correlations along the  $\langle 1100 \rangle$  direction persist. Finally, at  $T = 900$  K, long-range order is no longer apparent (Fig. 2.4.10(d)), in agreement with the near-zero SRO values in this high-temperature regime.

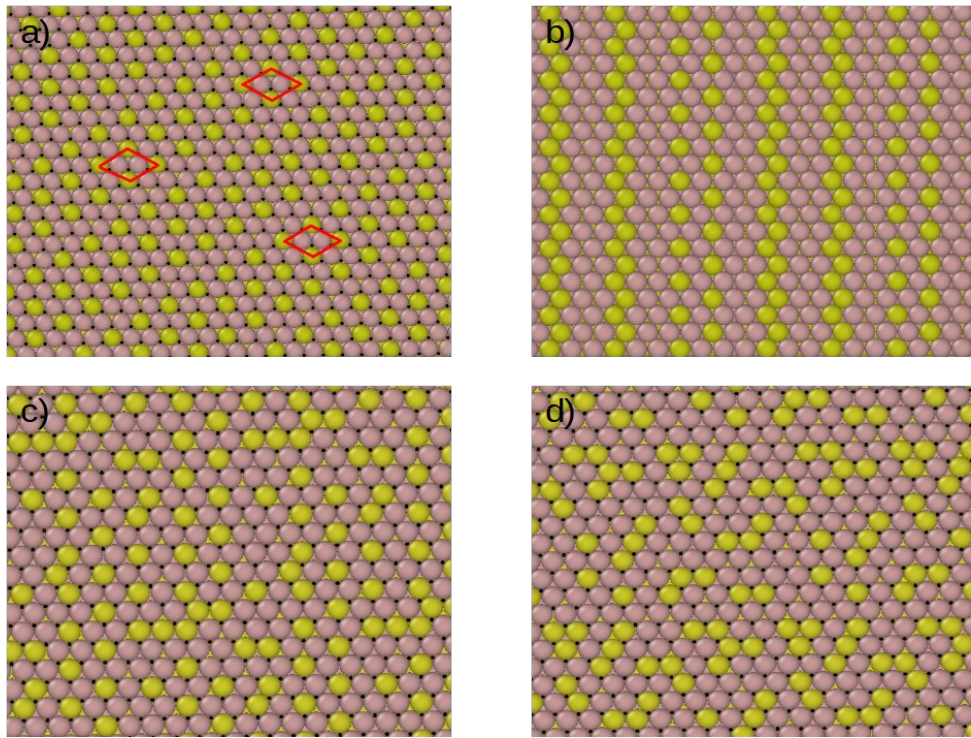


Figure 2.4.10 : Atomic configurations for  $\text{In}_x\text{Ga}_{1-x}\text{N}$  at composition  $x = 0.30$  at various temperatures. (a) and (b) correspond to  $T = 300$  K. (c) Corresponds to  $T = 700$  K, just above the order-disorder transition. (d) Configuration at  $T = 900$  K

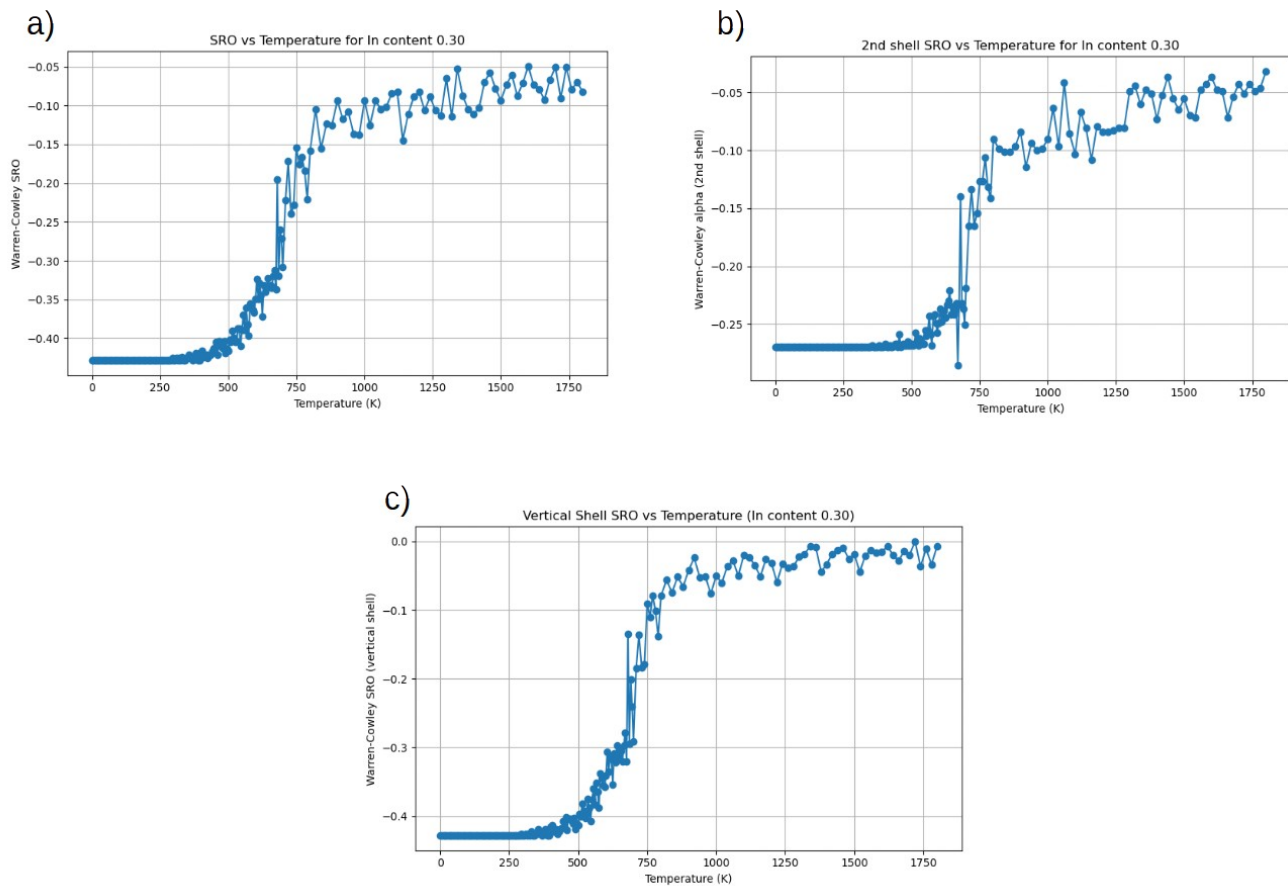


Figure 2.4.11 — Warren–Cowley short-range order (SRO) parameters for  $\text{In}_x\text{Ga}_{1-x}\text{N}$  at  $x = 0.30$  plotted as a function of temperature. (a)  $\alpha_a$ . (b)  $\alpha_m$ . (c)  $\alpha_c$ . All curves show a transition between 600–750 K, in agreement with the critical temperature ( $T_c \approx 690$  K) extracted from heat capacity data.

### $x=0.35$

At composition  $x = 0.35$ , the Warren–Cowley (SRO) parameters reveal a marked transition within the temperature range of 650–750 K, indicating the progressive loss of order as the system is thermally excited. This transition aligns well with the critical temperature determined from the heat capacity peak, which is approximately  $T_c=710$  K (see Fig. 2.2.4). The three SRO parameters plotted in Fig. 2.4.13 show strongly negative values near  $-0.35$  to  $-0.40$  at low temperatures, characteristic of pronounced ordering, and gradually increase toward zero as temperature rises. Atomic configurations illustrating this structural evolution are displayed in Fig. 2.4.10. At 300 K [Fig. 2.4.12(a),(b)], the In atoms demonstrate a clear tendency to occupy second-nearest neighbor positions both in the basal plane and along the  $c$ -axis, forming regular, strain-minimizing arrangements ( $\sqrt{3}\times\sqrt{3}$  patterns). At 900 K, well above the transition  $T$  [Fig. 2.4.12(d)], the ordering is lost and the system appears fully disordered, consistent with the nearly vanishing SRO values at this temperature. The intermediate configuration at 710 K [Fig. 2.4.12(c)], just above  $T_c$ , exhibits a partially disordered motif where remnant second-neighbor correlations persist.

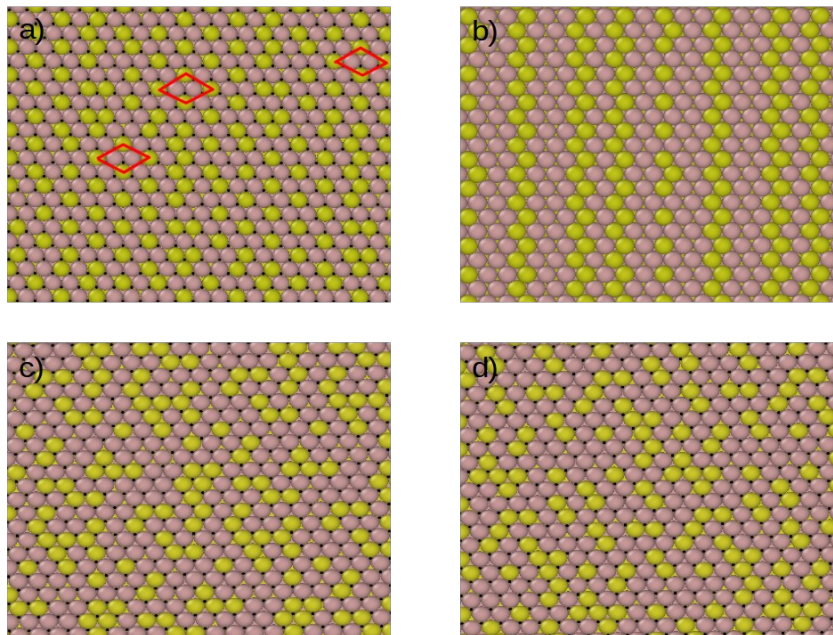


Figure 2.4.12: Atomic configurations of the  $\text{In}_x\text{Ga}_{1-x}\text{N}$  alloy for composition  $x = 0.35$  at various temperatures. (a) and (b) Top views at  $T = 300$  K, where In atoms are arranged predominantly as second-nearest neighbors both in the basal plane and along the  $[1100]$  direction, forming ordered

configurations. (c) Structure at  $T = 710$  K, just above the critical temperature. (d) Structure at  $T = 900$  K, where the distribution appears fully disordered, consistent with the loss of short-range order. (a), (c), and (d) is a top view of the basal plane, and (b) is a side view. Yellow balls are In atoms and brownish balls are Ga atoms.

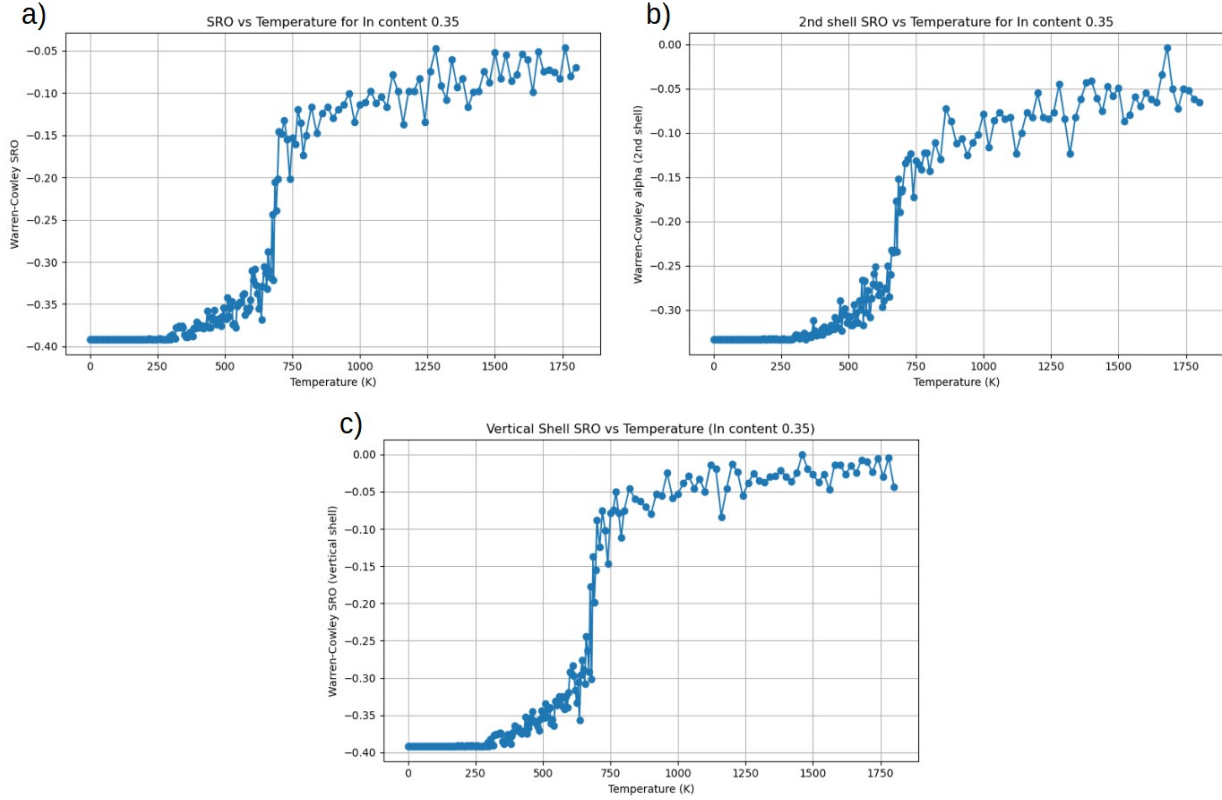


Figure 2.4.13: Warren–Cowley SRO parameters as a function of temperature for composition  $x = 0.35$ . (a)  $\alpha_a$ , (b)  $\alpha_m$ , and (c)  $\alpha_c$ . The transition from ordered to disordered state occurs between 650 K and 750 K, in agreement with the critical temperature  $T_c = 710$  K extracted from heat capacity.

### $x=0.40$

At composition  $x = 0.40$ , the Warren–Cowley SRO parameters exhibit a distinct transition between 650–700 K, consistent with the critical temperature obtained from the heat capacity peak, located at  $T_c = 670$  K (see Fig. 2.2.4). The three SRO parameters plotted in Fig. 2.4.15 start at approximately  $-0.35$  to  $-0.40$  at low temperatures and gradually approach zero as temperature increases, signaling a transition from an ordered to a disordered state. Corresponding atomic configurations are shown in Fig. 2.4.14. At  $T = 300$  K, the In atoms are preferentially aligned as second-nearest neighbor pairs within the basal plane, as seen in panels (a). At  $T = 740$  K, i.e., clearly above  $T_c$ , the arrangement becomes random (see Fig. 2.4.14(d)), consistent with the near-zero SRO values in this regime. At  $T = 680$  K, just above the critical temperature, partial degradation of the ordered structure is visible (Fig. 2.4.14(c)).



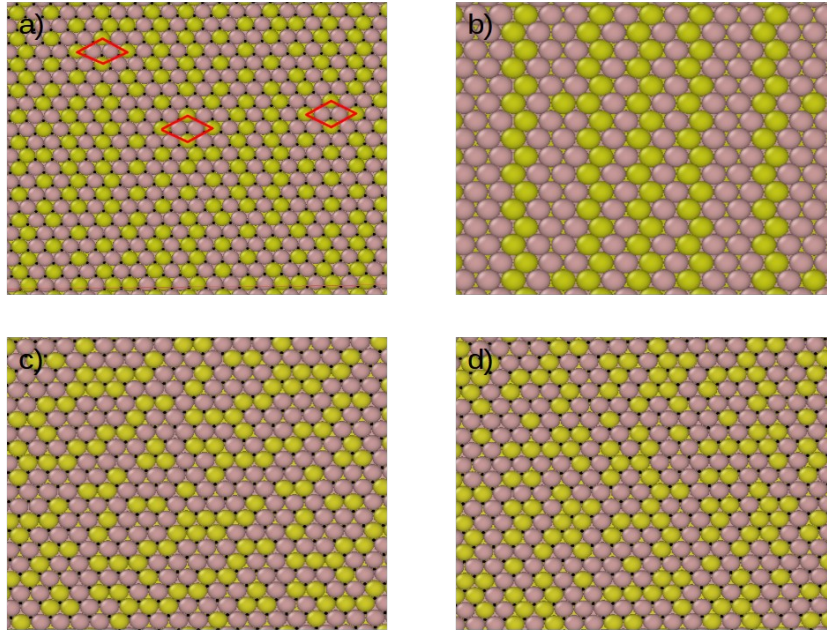


Figure 2.4.14: Atomic configurations of the  $\text{In}_x\text{Ga}_{1-x}\text{N}$  alloy for composition  $x = 0.40$  at various temperatures. (a) and (b) Top views at  $T = 300$  K, where In atoms are arranged predominantly as second-nearest neighbors both in the basal plane and along the  $\langle 0001 \rangle$  direction, forming ordered configurations. (c) Structure at  $T = 680$  K, just above the critical temperature. (d) Structure at  $T = 880$  K, where the distribution appears fully disordered, consistent with the loss of short-range order. (a), (c), and (d) is a top view of the basal plane, and (b) is a side view. Yellow balls are In atoms and brownish balls are Ga atoms.

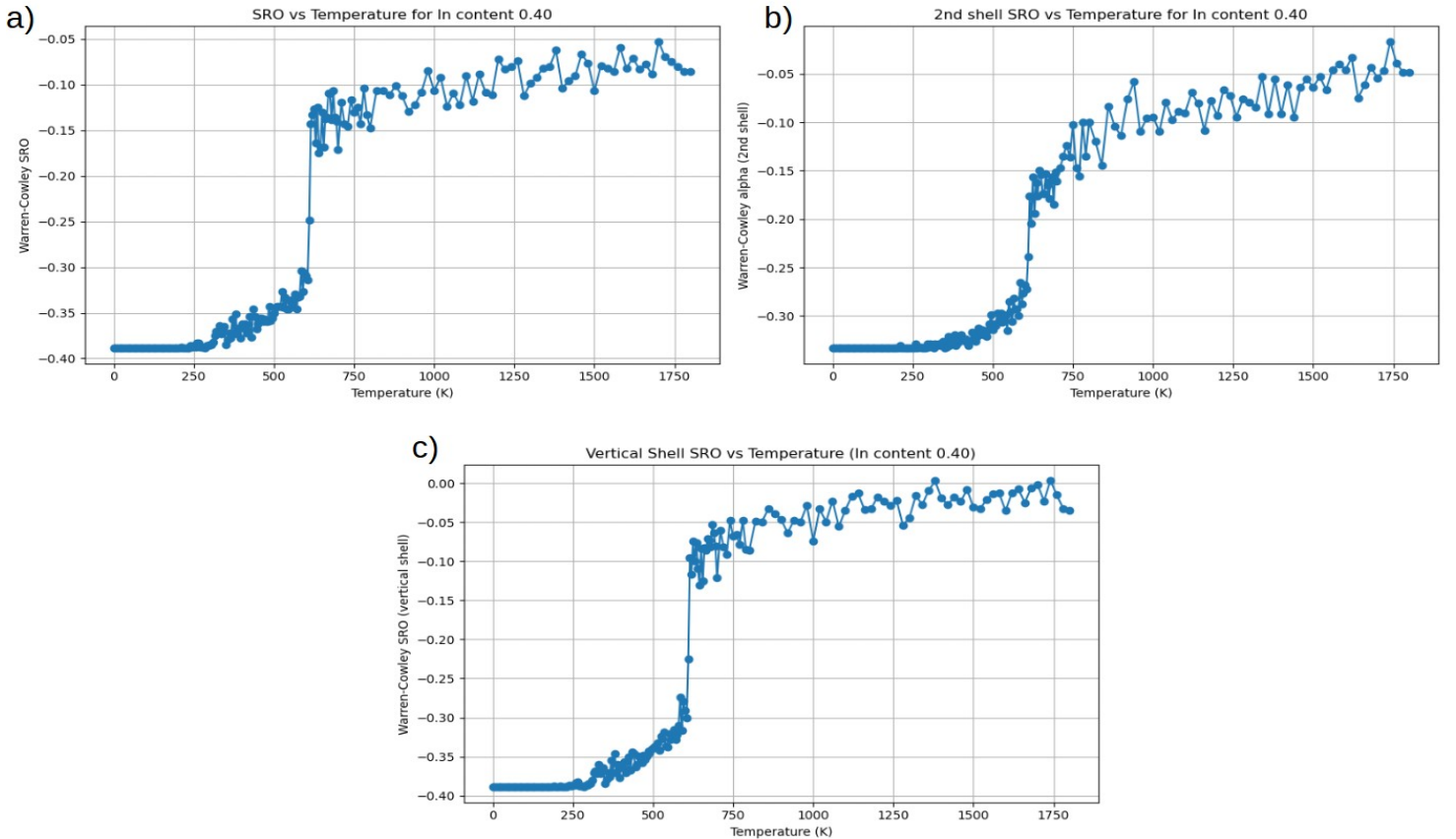


Figure 2.4.15: Warren–Cowley SRO parameters as a function of temperature for composition  $x = 0.40$ . (a)  $\alpha_a$ , (b)  $\alpha_m$ , and (c)  $\alpha_c$ . The transition from ordered to disordered state occurs between 650 K and 700 K, in agreement with the critical temperature  $T_c=660$  K extracted from heat capacity.

### Conclusions

In this thesis, we have studied the bulk alloy thermodynamics of  $\text{In}_x\text{Ga}_{1-x}\text{N}$  alloys by employing Monte Carlo calculations using a Cluster Expansion Hamiltonian. The analysis spanned a wide composition range from  $x=0.05$  to  $x=0.95$ , and temperatures from 0 K to 1800 K, providing a comprehensive picture of mixing energetics, phase separation tendencies, and ordering phenomena in the alloy.

The CE Hamiltonian was fitted to 160 DFT calculation InGaN structures (in a previous study) and can describe the energetics of these structures with an accuracy better than 4 meV/atom for the compositional range relevant for optoelectronic applications. These calculations assumed pseudomorphic InGaN alloys on GaN, i.e., biaxially strained to GaN. Interestingly, the convex hull of the mixing energy at  $T=0$  shows a single global minimum, indicating that decomposition towards the end constituents, i.e., InN and GaN, is suppressed for biaxially strained alloys.

Using the aforementioned CE Hamiltonian extensive MC calculations were conducted. Based on the mixing energies derived by these calculations a phase diagram has been constructed. This diagram indicates a rich space of different configurations and order-disorder transitions in the range of 600 - 750 K for compositions in the range  $x=0.2-0.4$

In addition to thermodynamic quantities, we investigated the microscopic atomic ordering in the alloy by analyzing Warren–Cowley short-range order (SRO) parameters and real-space atomic configurations. At low temperatures, we observed strong composition-dependent ordering motifs, driven by **strain-mediated In–In interactions**. Specifically, first-neighbor In–In repulsion and lack of these interaction at second nearest sites result in distinct patterns:  $\sqrt{3}\times\sqrt{3}$ -type in-plane ordering and vertical second-neighbor stacking. These motifs are most pronounced at intermediate compositions (e.g.,  $x=0.25$  to  $0.35$ ), where the system can most effectively relieve strain through spatial separation of In atoms.

A major outcome of this work is the detailed mapping of the **order–disorder transition** for each composition. The critical temperatures were also identified from peaks in the heat capacity data and further supported by the temperature dependence of the SRO parameters. Below  $T_c$ , the alloy displays clear short-range order, while above  $T_c$ , these correlations rapidly vanish as the system enters an entropically dominated disordered phase. Configuration snapshots across temperature show how ordered motifs are gradually disrupted near  $T_c$  and eventually lost at high temperatures.

A key outcome of the present work is that although phase separation towards GaN and InN is suppressed for pseudomorphic growth, at around 35% In content there is thermodynamic driving force for separation towards configurations with lower (approx. 25%) and higher (approx 40%) In contents. Nevertheless, it should be noted that for these areas of decomposition appear at rather low T and it is an open question whether they can experimentally be observed. However, this goes beyond the scope of this thesis and it is a topic for further investigations.



## Appendix A: Computational Scripts

Appendix A contains the Python scripts developed for this thesis. These scripts were used to perform data analysis, carry out Monte Carlo simulations, extract thermodynamic properties, and visualize atomic configurations and short-range order parameters.

### A.0 DFT Data Processing to Database

This script parses a custom DFT output file (structures.in), extracts lattice vectors, atomic positions (in direct coordinates), total energies, and atom counts per species, and then stores the data into an ASE file .db with computed mixing energies and compositions.

```
from ase import Atoms

from ase.db import connect

import numpy as np


def read_data_from_file(file_path):

    prim_vecs = []
    direct_cords = []
    energies = []
    species = []

    with open(file_path, 'r') as file:

        lines = file.readlines()

    for i in range(len(lines)):

        line = lines[i].strip()
```

```

if line == 'Direct':
    # Extract atomic positions in direct coordinates
    direct_cord = []
    j = i + 1
    line = lines[j].strip()
    while len(line.split()) == 3:
        cord = tuple(map(float, line.split()))
        direct_cord.append(cord)
        j += 1
        line = lines[j].strip()
    direct_cords.append(direct_cord)

    # Extract primitive lattice vectors
    prim_vec = []
    k = i - 4
    for _ in range(3):
        vec = tuple(map(float, lines[k].strip().split()))
        prim_vec.append(vec)
        k += 1
    prim_vecs.append(prim_vec)

    # Extract total energy
    energies.append(float(lines[j].strip()))

    # Extract species counts (e.g., number of In and Ga atoms)
    species_line = list(map(int, lines[i - 1].strip().split()))
    species.append(species_line)

```

```

return prim_vecs, direct_cords, energies, species

# Input DFT data
input_file_path = 'structures.in'

# Parse data
prim_vecs, direct_cords, energies, species =
read_data_from_file(input_file_path)

# Create ASE database
db = connect('new.db')

# Write data to database
for i in range(len(prim_vecs)):
    lattice_vectors = np.array(prim_vecs[i]).round(6)
    atomic_positions = np.dot(np.array(direct_cords[i]),
lattice_vectors).round(15)

    sp = ['In'] * species[i][0] + ['Ga'] * species[i][1]
    atoms = Atoms(sp, positions=atomic_positions, cell=lattice_vectors,
pbc=True)

# Energies of pure binaries (per atom)
e_tot_GaN = -27.882691 / 2
e_tot_InN = -23.514267 / 2

# Mixing energy per atom
mixing_energy = (
    energies[i] - species[i][0] * e_tot_InN - species[i][1] * e_tot_GaN

```

```
) / (species[i][0] + species[i][1])

concentration = species[i][0] / (species[i][0] + species[i][1])

tag_string = ".join(sp)

db.write(atoms, tag=tag_string, mixing_energy=mixing_energy,
concentration=concentration, lattice_parameter=1.0)
```

## A.1: Cluster Cutoff Optimization

Before constructing the final Cluster Expansion model, a systematic search was performed to determine appropriate cutoff radii for pairs, triplets, and quadruplets. This script trains models at various cutoff distances and evaluates their performance using cross-validation. The Root Mean Squared Error (RMSE), BIC, and number of parameters were used as selection metrics.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from ase.db import connect
from icet import ClusterSpace, StructureContainer
from trainstation import CrossValidationEstimator

# Define how to create the structure container and get fit data
def get_fit_data(cutoffs):
    cs = ClusterSpace(position_tolerance=1e-03,
                      structure=primitive_structure,
                      cutoffs=cutoffs,
                      chemical_symbols=['In', 'Ga'])
    sc = StructureContainer(cluster_space=cs)
    for row in db.select():
        sc.add_structure(structure=row.toatoms(),
                        user_tag=row.tag,
                        properties={'mixing_energy': row.mixing_energy})
    return sc.get_fit_data(key='mixing_energy')

# Train CE with cross-validation and return fit statistics
def train_ce(cutoffs):
    A, y = get_fit_data(cutoffs)
    cve = CrossValidationEstimator((A, y), fit_method=fit_method,
                                  validation_method='shuffle-split',
                                  n_splits=100)

    cve.validate()
    cve.train()
```

```

    return {
        'rmse_validation': cve.rmse_validation,
        'rmse_train': cve.rmse_train,
        'BIC': cve.model.BIC,
        'n_parameters': cve.n_parameters,
        'n_nonzero_parameters': cve.n_nonzero_parameters
    }

# Setup
fit_method = 'lasso'
db = connect('new.db')
primitive_structure = db.get(id=1).toatoms()

# Define cutoff ranges
c2_vals = np.arange(0.0, 3.0, 0.1)
c3_vals = np.arange(0.0, 3.0, 0.5)
c4_vals = np.arange(0.0, 1.5, 0.1)

# --- Step 1: Optimize 2nd-order cutoff ---
records = []
for c2 in c2_vals:
    row = train_ce([c2])
    records.append({'c2': c2, **row})
df2 = pd.DataFrame(records)
c2_final = 3.0 # Chosen based on RMSE/BIC analysis

# Plot 2nd-order results
fig, axes = plt.subplots(figsize=(4, 5.2), dpi=120, sharex=True, nrows=3)
axes[0].plot(df2.c2, 1000 * df2.rmse_validation, '-o', label='validation')
axes[0].plot(df2.c2, 1000 * df2.rmse_train, '--s', label='train')
axes[0].set_ylabel('RMSE (meV/atom)')
axes[0].legend()

axes[1].plot(df2.c2, df2.BIC * 1e-3, '-o')
axes[1].set_ylabel(r'BIC ( $\times 10^3$ )')

axes[2].plot(df2.c2, df2.n_parameters, '--s', label='Total')
axes[2].plot(df2.c2, df2.n_nonzero_parameters, '-o', label='Nonzero')
axes[2].set_xlabel('2nd order cutoff (Å)')
axes[2].set_ylabel('Number of parameters')
axes[2].legend()
fig.tight_layout()

# --- Step 2: Optimize 3rd-order cutoff ---
records = []
for c3 in c3_vals:
    row = train_ce([c2_final, c3])
    records.append({'c2': c2_final, 'c3': c3, **row})

```

```

df3 = pd.DataFrame(records)
c3_final = 2.0

# Plot 3rd-order results
fig, axes = plt.subplots(figsize=(4, 5.2), dpi=120, sharex=True, nrows=3)
axes[0].plot(df3.c3, 1000 * df3.rmse_validation, '-o', label='validation')
axes[0].plot(df3.c3, 1000 * df3.rmse_train, '--s', label='train')
axes[0].set_ylabel('RMSE (meV/atom)')
axes[0].legend()

axes[1].plot(df3.c3, df3.BIC * 1e-3, '-o')
axes[1].set_ylabel(r'BIC ($\times 10^{\{3\}}$)')

axes[2].plot(df3.c3, df3.n_parameters, '--s', label='Total')
axes[2].plot(df3.c3, df3.n_nonzero_parameters, '-o', label='Nonzero')
axes[2].set_xlabel('3rd order cutoff (Å)')
axes[2].set_ylabel('Number of parameters')
axes[2].legend()
fig.tight_layout()

# --- Step 3: Optimize 4th-order cutoff ---
records = []
for c4 in c4_vals:
    row = train_ce([c2_final, c3_final, c4])
    records.append({'c2': c2_final, 'c3': c3_final, 'c4': c4, **row})
df4 = pd.DataFrame(records)
c4_final = 1.5

# Plot 4th-order results
fig, axes = plt.subplots(figsize=(4, 5.2), dpi=120, sharex=True, nrows=3)
axes[0].plot(df4.c4, 1000 * df4.rmse_validation, '-o', label='validation')
axes[0].plot(df4.c4, 1000 * df4.rmse_train, '--s', label='train')
axes[0].set_ylabel('RMSE (meV/atom)')
axes[0].legend()

axes[1].plot(df4.c4, df4.BIC * 1e-3, '-o')
axes[1].set_ylabel(r'BIC ($\times 10^{\{3\}}$)')

axes[2].plot(df4.c4, df4.n_parameters, '--s', label='Total')
axes[2].plot(df4.c4, df4.n_nonzero_parameters, '-o', label='Nonzero')
axes[2].set_xlabel('4th order cutoff (Å)')
axes[2].set_ylabel('Number of parameters')
axes[2].legend()
fig.tight_layout()

```

## A.2: Cluster Expansion Fitting and Validation

This script builds the Cluster Expansion (CE) model using the ICET package. The model is trained on DFT-derived mixing energies and validated using ARD Regression. Additional steps include prediction, accuracy plots, and convex hull construction.

```
from ase.db import connect
from icet import ClusterSpace, StructureContainer, ClusterExpansion
from trainstation import CrossValidationEstimator
import matplotlib.pyplot as plt
from icet.tools import ConvexHull, enumerate_structures
import numpy as np
import pandas as pd

# Load the primitive structure from the ASE database
db = connect('new.db')
primitive_structure = db.get(id=1).toatoms()

# Define cluster space with optimized cutoffs
cs = ClusterSpace(position_tolerance=1e-03,
                  structure=primitive_structure,
                  cutoffs=[3.00, 2.00, 1.50],
                  chemical_symbols=['In', 'Ga'])
print(cs)

# Create and populate structure container with DFT data
sc = StructureContainer(cluster_space=cs)
for row in db.select():
    structure = row.toatoms()
    try:
        sc.add_structure(structure=structure,
                        user_tag=row.tag,
                        properties={'mixing_energy': row.mixing_energy})
    except Exception as e:
        print(f"Error adding structure for tag {row.tag}: {e}")

print(sc)

# Train the CE model using ARD Regression
opt =
CrossValidationEstimator(fit_data=sc.get_fit_data(key='mixing_energy', fit_
method='ardr')
opt.validate()
opt.train()
print(opt)
```

```
# Build the final Cluster Expansion object and save it
ce = ClusterExpansion(cluster_space=cs,
                      parameters=opt.parameters,
                      metadata=opt.summary)
print(ce)
ce.write('mixing_energy.ce')
```

### Model Validation and Energy Prediction

```
# Load the CE model and compare predictions with DFT values
ce = ClusterExpansion.read('mixing_energy.ce')
data = {'concentration': [], 'reference_energy': [], 'predicted_energy': [],
        'deviation': []}
db = connect('new.db')

for row in db.select('natoms<=1000'):
    concentration = row.concentration
    ref = 1e3 * row.mixing_energy
    pred = 1e3 * ce.predict(row.toatoms())
    deviation = abs((pred - ref) / pred) * 100

    data['concentration'].append(concentration)
    data['reference_energy'].append(ref)
    data['predicted_energy'].append(pred)
    data['deviation'].append(deviation)

# Plot CE prediction vs. reference DFT mixing energies
fig, ax = plt.subplots(figsize=(4, 3))
ax.set_xlabel(r'Ga concentration')
ax.set_ylabel(r'Mixing energy (meV/atom)')
ax.set_xlim([0, 1])
ax.set_ylim([-150, 50])
ax.scatter(data['concentration'], data['reference_energy'], marker='o',
           label='reference')
ax.scatter(data['concentration'], data['predicted_energy'], marker='x',
           label='CE prediction')
plt.legend()
plt.savefig('mixing_energy_comparison.png', bbox_inches='tight')
```

### Convex Hull Construction from Enumerated Structures

```
# Predict CE energies for enumerated structures

ce = ClusterExpansion.read('mixing_energy.ce')

cluster_space = ce.get_cluster_space_copy()

primitive_structure = cluster_space.primitive_structure
```



```
chemical_symbols = cluster_space.chemical_symbols

data = {'concentration': [], 'mixing_energy': []}
structures = []
for structure in enumerate_structures(position_tolerance=1e-03,
                                     structure=primitive_structure,
                                     sizes=range(1, 7),
                                     chemical_symbols=chemical_symbols):
    conc = structure.get_chemical_symbols().count('Ga') / len(structure)
    energy = ce.predict(structure)
    data['concentration'].append(conc)
    data['mixing_energy'].append(energy)
    structures.append(structure)

print(f'Predicted energies for {len(structures)} structures')

# Construct and plot convex hull
hull = ConvexHull(data['concentration'], data['mixing_energy'])

fig, ax = plt.subplots(figsize=(4, 3))
ax.set_xlabel(r'Ga concentration')
ax.set_ylabel(r'Mixing energy (meV/atom)')
ax.set_xlim([0, 1])
ax.set_ylim([-110, 15])
ax.scatter(data['concentration'], 1e3 * np.array(data['mixing_energy']),
           marker='x')
ax.plot(hull.concentrations, 1e3 * hull.energies, '-o', color='green')
plt.savefig('mixing_energy_predicted.png', bbox_inches='tight')
```

### A.3: Monte Carlo Simulation of Thermodynamic Properties

This script runs canonical ensemble Monte Carlo simulations on a CE-derived Hamiltonian. For each temperature, the internal energy and heat capacity are computed from the **last 500,000 steps** of the simulation to ensure equilibrium. Structures and thermodynamic data are saved per temperature.

**Note:** If continuing from a past simulation, leave the structure loading section as-is. If starting fresh, comment it out and use random structure generation instead.

```
from mchammer.ensembles import CanonicalEnsemble

from mchammer.calculators import ClusterExpansionCalculator

from ase.build import make_supercell

from icet import ClusterExpansion

import numpy as np

from ase.db import connect

from icet.tools.structure_generation import occupy_structure_randomly

from icet import ClusterSpace, StructureContainer

import matplotlib.pyplot as plt

from mchammer import DataContainer

from ase.io import write, read

import time

import os

start_time = time.perf_counter()

kB = 8.617333 262e-5 # Boltzmann constant in eV/K
```

```
def compute_mean_energy_and_heat_capacity(dc, N, T,
last_n_steps=500 000):

    """Computes mean energy and heat capacity from the last N steps."""

    energies = dc.get('potential')[-last_n_steps:] / N

    mean_energy = np.mean(energies)

    mean_energy_sq = np.mean(energies**2)

    heat_capacity = (mean_energy_sq - mean_energy**2) / (kB * T**2)

    return mean_energy, heat_capacity


def run_mc_fixed_steps(supercell, calc, T, xyz_fname, N,
n_steps=1 000 000):

    """Runs fixed-step Monte Carlo and computes observables from last
steps."""

    print(f"Running MC at {T}K for {n_steps} steps")

    mc = CanonicalEnsemble(supercell, calc, T,
ensemble_data_write_interval=1)

    mc.run(n_steps)

    dc = mc.data_container

    supercell = mc.structure.copy()

    mean_energy, heat_capacity =
compute_mean_energy_and_heat_capacity(dc, N, T)

    print(f"Mean Energy: {mean_energy:.6f} eV, Heat Capacity:
{heat_capacity:.8f} eV/K")

    write(xyz_fname, mc.structure)
```

```
return supercell, mean_energy, heat_capacity

# Load cluster expansion and prepare supercell
ce = ClusterExpansion.read('mixing_energy.ce')
primitive_structure = ce.primitive_structure

# Use this section to start from an existing structure
supercell = read('1000/0.95/final_structure_1400K.xyz')

# Alternatively, comment the line above and uncomment below to start
fresh

# supercell = occupy_structure_randomly(primitive_structure,
#
#                                     supercell_size=[10, 10, 5],
#
#                                     symbols=['In', 'Ga'])

N = len(supercell)

calc = ClusterExpansionCalculator(supercell, ce)

temperatures = np.arange(1380, 1000, -20)

file_path = "1000/0.95/output.txt"

# Skip already computed temperatures
computed_temperatures = set()

if os.path.exists(file_path):
    with open(file_path, "r") as f:
```

```
for line in f:

    try:

        T_done = int(line.split()[0])

        computed_temperatures.add(T_done)

    except (IndexError, ValueError):

        continue


# Run simulations

for T in map(int, temperatures):

    if T in computed_temperatures:

        print(f"Skipping T = {T}K (already computed)")

        continue


    xyz_fname = f'1000/0.95/final_structure_{T}K.xyz'

    supercell, mean_energy, heat_capacity = run_mc_fixed_steps(supercell,
calc, T, xyz_fname, N)


    with open(file_path, "a") as f:

        f.write(f"{T} {mean_energy} {heat_capacity}\n")

        f.flush()


print("Data saved to output.txt")

end_time = time.perf_counter()

print(f"Elapsed time : {end_time - start_time:.2f} sec")
```

## A.4: Phase Diagram – Chemical Potential Difference $\Delta\mu$

This script computes the chemical potential difference

$\Delta\mu = \mu_{\text{In}} - \mu_{\text{Ga}}$  as a function of composition and temperature using mixing energies from Monte Carlo simulations. The results are visualized as a contour phase diagram. The end points  $x = 0.0$  and  $x = 1.0$  are both treated as reference values with  $E_{\text{mix}}(x=0,T)=0$

```
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy.interpolate import griddata
from scipy.ndimage import gaussian_filter
import seaborn as sns

# === Constants ===
EInN = -11.7571335
EGaN = -13.9413455
delta_x = 0.05
base_dir = 'final' # Folder with subfolders like '0.05', '0.10', ..., '0.95'

# === Read E_mix(x, T) ===
data = []
for folder in os.listdir(base_dir):
    folder_path = os.path.join(base_dir, folder)
    if not os.path.isdir(folder_path):
        continue
    try:
        x = float(folder)
    except ValueError:
        continue

    output_file = os.path.join(folder_path, 'output.txt')
    if not os.path.isfile(output_file):
        continue

    df_temp = pd.read_csv(output_file, delim_whitespace=True,
                           header=None, names=["T", "Emix", "Cp"])
    for _, row in df_temp.iterrows():
        data.append([x, row["T"], row["Emix"]])

df = pd.DataFrame(data, columns=["x", "T", "Emix"])

# Add x = 0.0 and x = 1.0 as pure endpoints with Emix = 0.0
```

```

unique_temps = df["T"].unique()
df_extra = pd.DataFrame({
    "x": [0.0] * len(unique_temps) + [1.0] * len(unique_temps),
    "T": list(unique_temps) * 2,
    "Emix": 0.0
})
df = pd.concat([df, df_extra], ignore_index=True)

# === Compute  $\Delta\mu$ (In-Ga) ===
dmu_data = []
for T in sorted(unique_temps):
    df_T = df[df["T"] == T].groupby("x", as_index=True)["Emix"].mean()

    for x in np.arange(0.05, 1.01, 0.05):
        x = round(x, 2)
        x_prev = round(x - delta_x, 2)
        if x in df_T.index and x_prev in df_T.index:
            Emix_x = float(df_T.loc[x])
            Emix_prev = float(df_T.loc[x_prev])
            dEmix_dx = (Emix_x - Emix_prev) / delta_x
            dmu = 2 * dEmix_dx + 2 * (EInN - EGaN)
            dmu_data.append([x, T, dmu])

# Add  $\Delta\mu$  at x = 0.0 and x = 1.0 manually
for T in sorted(unique_temps):
    dmu_edge = 2 * (EInN - EGaN)
    dmu_data.extend([
        [0.0, T, dmu_edge],
        [1.0, T, dmu_edge]
    ])

# Create  $\Delta\mu$  DataFrame
dmu_df = pd.DataFrame(dmu_data, columns=["x", "T", "dmu"])

# === Interpolate  $\Delta\mu$  for contour plotting ===
x_vals = dmu_df["x"].to_numpy()
T_vals = dmu_df["T"].to_numpy()
dmu_vals = dmu_df["dmu"].to_numpy()

x_grid = np.linspace(0.0, 1.0, 200)
T_grid = np.linspace(T_vals.min(), T_vals.max(), 200)
x_mesh, T_mesh = np.meshgrid(x_grid, T_grid)

dmu_grid = griddata((x_vals, T_vals), dmu_vals, (x_mesh, T_mesh),
    method="linear")
dmu_grid_smoothed = gaussian_filter(dmu_grid, sigma=1.5)

# === Plot  $\Delta\mu$  phase diagram ===

```

```
sns.set_context("talk", font_scale=1.1)
sns.set_style("whitegrid")
plt.figure(figsize=(14, 7))

# Filled contour
contourf_obj = plt.contourf(
    x_mesh, T_mesh, dmu_grid_smoothed,
    levels=100, cmap="plasma"
)

# Black lines
contour_lines = plt.contour(
    x_mesh, T_mesh, dmu_grid_smoothed,
    levels=100, colors="black", linewidths=0.6
)

# Colorbar
cbar = plt.colorbar(contourf_obj)
cbar.set_label(r"$\Delta \mu(\mathrm{In-Ga})$", fontsize=12)

# Labels and formatting
plt.xlabel(r'Concentration  $x$  in  $\mathrm{In}_x\mathrm{Ga}_{1-x}\mathrm{N}$ ', fontsize=13)
plt.ylabel('Temperature (K)', fontsize=13)
plt.title(r'$\Delta \mu(\mathrm{In-Ga})$ Phase Diagram', fontsize=15)
xticks = np.arange(0.0, 1.01, 0.05)
plt.xticks(xticks, [f"{x:.2f}" for x in xticks])
plt.yticks(np.linspace(T_vals.min(), T_vals.max(), 7))
plt.grid(True, linestyle='--', alpha=0.5)
plt.tight_layout()
plt.show()
```

## A.5: Thermodynamic Analysis — Entropy and Free Energy of Mixing

This script calculates the temperature-dependent mixing entropy and Helmholtz free energy using canonical Monte Carlo results for  $E_{\text{mix}}(x, T)$ . The entropy is derived via thermodynamic integration and plotted alongside the ideal entropy. Results are saved in designated folders per temperature.

```
import os
import numpy as np
import matplotlib.pyplot as plt
from collections import defaultdict
from scipy.integrate import cumtrapz

# === Configuration ===
```



```

BASE_DIR = "final1"                # Folder with composition subfolders
(e.g. '0.05', '0.10', ...)
EMIX_FILENAME = "output.txt"       # File containing T, E_mix, Cp
RESULT_DIR = "results"             # Root result directory
ENTROPY_DIR = os.path.join(RESULT_DIR, "entropy_plots")
FREE_ENERGY_DIR = os.path.join(RESULT_DIR, "free_energy_plots")
kB = 8.617333 262 145e-5           # Boltzmann constant in eV/K

# === Create output folders ===
os.makedirs(ENTROPY_DIR, exist_ok=True)
os.makedirs(FREE_ENERGY_DIR, exist_ok=True)

# === Parse E_mix(T) for each composition ===
entropy_data = defaultdict(list)
free_energy_data = defaultdict(list)

for folder in sorted(os.listdir(BASE_DIR), key=lambda x: float(x)):
    try:
        x = float(folder)
    except ValueError:
        continue

    path = os.path.join(BASE_DIR, folder, EMIX_FILENAME)
    if not os.path.exists(path):
        continue

    data = np.loadtxt(path)
    if data.ndim == 1:
        data = data.reshape(1, -1)

    sort_idx = np.argsort(data[:, 0])
    T = data[sort_idx, 0]
    E = data[sort_idx, 1]

    # === Thermodynamic Integration ===
    F_ref = E[0]                    # Free energy reference point at lowest temperature
    T_ref = T[0]
    integral = cumtrapz(E / T**2, T, initial=0)
    F_mix = T * (F_ref / T_ref - integral)
    S_mix = -np.gradient(F_mix, T) / kB

    for temp, S_val, F_val in zip(T, S_mix, F_mix):
        entropy_data[temp].append((x, S_val))
        free_energy_data[temp].append((x, F_val))

# === Plot S_mix(x, T) vs Ideal Entropy ===
for T in sorted(entropy_data.keys()):
    data = sorted(entropy_data[T])

```

```

x_vals = np.array([x for x, S in data])
S_vals = np.array([S for x, S in data])

# Ideal entropy curve:  $S = -x \log x - (1-x) \log(1-x)$ 
x_ideal = np.linspace(0.001, 0.999, 300)
S_ideal = -x_ideal * np.log(x_ideal) - (1 - x_ideal) * np.log(1 - x_ideal)

plt.figure(figsize=(8, 6))
plt.plot(x_vals, S_vals, 'o-', label="MC entropy")
plt.plot(x_ideal, S_ideal, 'k--', linewidth=2, label="Ideal entropy")
plt.xlabel("In composition x")
plt.ylabel(r"$S_{\mathrm{mix}}(x, T)$ (k_B$/atom)")
plt.title(f"Mixing Entropy at T = {int(T)} K")
plt.grid(True)
plt.legend()
plt.tight_layout()

plot_path = os.path.join(ENTROPY_DIR, f"entropy_T{int(T)}K.png")
plt.savefig(plot_path)
plt.close()

# === Plot F_mix(x, T) ===
for T in sorted(free_energy_data.keys()):
    data = sorted(free_energy_data[T])
    x_vals = np.array([x for x, F in data])
    F_vals = np.array([F for x, F in data])

    plt.figure(figsize=(8, 6))
    plt.plot(x_vals, F_vals, 'o-', color='tab:red', label="MC free energy")
    plt.xlabel("In composition x")
    plt.ylabel(r"$F_{\mathrm{mix}}(x, T)$ (eV/atom)")
    plt.title(f"Mixing Free Energy at T = {int(T)} K")
    plt.grid(True)
    plt.legend()
    plt.tight_layout()

    plot_path = os.path.join(FREE_ENERGY_DIR,
f"free_energy_T{int(T)}K.png")
    plt.savefig(plot_path)
    plt.close()

```

## A.6: Warren–Cowley Short-Range Order (SRO) Parameter Analysis

This appendix presents the methods and scripts used to compute the Warren–Cowley short-range order (SRO) parameters in the  $\text{In}_x\text{Ga}_{1-x}\text{N}$  alloy system using configurations generated from canonical Monte Carlo simulations. Both in-plane and out-of-plane neighbor interactions are considered to quantify atomic ordering across different coordination shells and spatial directions.

### A.6.1 In-Plane SRO Calculations

Short-range order parameters calculated using neighbors located within the same atomic z-plane

- **A.6.2.1 First-Shell In-Plane ( $\alpha_1$ )**  
Immediate neighbors within a radius cutoff in the same plane.
- **A.6.2.2 Second-Shell In-Plane ( $\alpha_2$ )**  
Next-nearest neighbors in a defined radial shell range in the same plane.
- **A.6.2.3 Third-Shell In-Plane ( $\alpha_3$ )**  
Third-neighbor shell using larger radial cutoffs, still confined to one plane.

### A.6.2 Out-of-Plane SRO Calculation

SRO parameters evaluated across neighboring z-planes to capture vertical or diagonal ordering.

- **A.6.3.1 Next-Plane Neighbors**  
Atoms in the plane directly above the reference layer, using a fixed plane\_gap and bond-radius threshold.
- **A.6.3.2 Plane-by-Plane Traversal**  
Generalized approach to find neighbors across multiple planes, with full z-PBC handling and optional second-plane search.

### Appendix A.6: Warren–Cowley Short-Range Order (SRO) Parameter Analysis

This appendix presents the scripts used to evaluate short-range order (SRO) parameters for the  $\text{In}_x\text{Ga}_{1-x}\text{N}$  alloy using atomic configurations from Monte Carlo simulations. These include calculations for both in-plane and out-of-plane atomic interactions across various neighbor shells.

# A.6.1 In-Plane SRO Calculations

### ## A.6.1.1 First-Shell In-Plane ( $\alpha_1$ )

```

from ase.io import read
import numpy as np

def find_in_plane_neighbors_with_pbc(atoms, bond_radius=1.0,
z_tolerance=0.2):
    neighbors_dict = {}
    positions = atoms.get_positions()
    for i, pos_i in enumerate(positions):
        neighbors = []
        for j, pos_j in enumerate(positions):
            if i != j and abs(pos_i[2] - pos_j[2]) < z_tolerance:
                dist = atoms.get_distance(i, j, mic=True)
                if dist <= bond_radius:
                    neighbors.append(j)
        neighbors_dict[i] = neighbors
    return neighbors_dict

```

### ## A.6.1.2 Second-Shell In-Plane ( $\alpha_2$ )

```

def find_in_plane_shell_neighbors_with_pbc(atoms, min_radius,
max_radius, z_tolerance=0.2):
    neighbors_dict = {}
    positions = atoms.get_positions()
    for i, pos_i in enumerate(positions):
        neighbors = []
        for j, pos_j in enumerate(positions):
            if i != j and abs(pos_i[2] - pos_j[2]) < z_tolerance:
                dist = atoms.get_distance(i, j, mic=True)
                if min_radius < dist <= max_radius:
                    neighbors.append(j)
        neighbors_dict[i] = neighbors
    return neighbors_dict

```

### ## A.6.1.3 Third-Shell In-Plane ( $\alpha_3$ )

```

def find_3rd_in_plane_neighbors_with_pbc(atoms, min_radius=3.0,
max_radius=3.5, z_tolerance=0.4):
    neighbors_dict = {}
    positions = atoms.get_positions()
    for i, pos_i in enumerate(positions):
        neighbors = []
        for j, pos_j in enumerate(positions):
            if i != j and abs(pos_i[2] - pos_j[2]) < z_tolerance:

```

```

        dist = atoms.get_distance(i, j, mic=True)
        if min_radius < dist <= max_radius:
            neighbors.append(j)
        neighbors_dict[i] = neighbors
    return neighbors_dict

```

#### # A.6.2 Out-of-Plane SRO Calculations

##### ## A.6.2.1 Next-Plane Neighbors ( $\alpha_{\{z+1\}}$ )

```

def find_neighbors_across_planes_with_pbc(atoms, bond_radius=1.0,
plane_gap=0.5, z_tolerance=0.2):
    neighbors_dict = {}
    positions = atoms.get_positions()
    for i, pos_i in enumerate(positions):
        z_i = pos_i[2]
        neighbors = []
        for j, pos_j in enumerate(positions):
            if i != j:
                dz = pos_j[2] - z_i
                if abs(dz - plane_gap) <= z_tolerance:
                    dist = atoms.get_distance(i, j, mic=True)
                    if dist <= bond_radius:
                        neighbors.append(j)
        neighbors_dict[i] = neighbors
    return neighbors_dict

```

##### ## A.6.2.2 Generalized Out-of-Plane Neighbors (Plane-by-Plane)

```

def find_neighbors_across_planes_with_pbc(
    atoms,
    min_bond_radius=1.90,
    max_bond_radius=1.92,
    plane_gap=1.63229,
    z_tolerance=0.2
):
    neighbors_dict = {}
    positions = atoms.get_positions()
    cell = atoms.get_cell()
    Lz = cell[2, 2] # Length of cell in z-direction
    for i, pos_i in enumerate(positions):
        z_i = pos_i[2]
        neighbor_candidates = []
        for j, pos_j in enumerate(positions):
            if i == j:
                continue

```

```

        dz = pos_j[2] - z_i
        dz_pbc = dz - Lz * round(dz / Lz)
        if (plane_gap - z_tolerance) <= dz_pbc <= (plane_gap + z_tolerance)
and dz_pbc > 0:
            dist = atoms.get_distance(i, j, mic=True)
            if min_bond_radius <= dist <= max_bond_radius:
                neighbor_candidates.append((j, dist))
            neighbors_dict[i] = neighbor_candidates
    return neighbors_dict

```

### # A.6.3 SRO Evaluation Function

```

def compute_sro_warren_cowley(atoms, neighbors, type_A='In',
type_B='Ga'):
    n_A = 0
    total_B_neighbors = 0
    total_neighbors = 0
    for i, atom in enumerate(atoms):
        if atom.symbol == type_A and i in neighbors:
            n_A += 1
            B_neighbors = sum(
                1 for j in neighbors[i]
                if (isinstance(j, int) and atoms[j].symbol == type_B)
                or (isinstance(j, tuple) and atoms[j[0]].symbol == type_B)
            )
            total_B_neighbors += B_neighbors
            total_neighbors += len(neighbors[i])
    if n_A == 0 or total_neighbors == 0:
        return None
    P_AB = total_B_neighbors / total_neighbors
    x_B = sum(1 for atom in atoms if atom.symbol == type_B) / len(atoms)
    return 1 - (P_AB / x_B)

```

### # Example usage:

```

atoms = read("final1/0.45/final_structure_860K.xyz")
neighbors = find_in_plane_neighbors_with_pbc(atoms)
alpha = compute_sro_warren_cowley(atoms, neighbors)
print(f"SRO  $\alpha_1 = \{\text{alpha:.4f}\}")$ 
```

## Appendix B : Regression Methods

### B.1 LASSO Regression

LASSO (Least Absolute Shrinkage and Selection Operator) is a linear regression method that performs both regularization and feature selection. It minimizes the residual sum of squares with an added penalty on the absolute values of the coefficients:

$$\min_{\beta} \left[ \frac{1}{2n} \sum_{i=1}^n (y_i - X_i \beta)^2 + \lambda \sum_{j=1}^p |\beta_j| \right]$$

Here,  $\lambda$  controls the strength of the penalty. As  $\lambda$  increases, more coefficients are driven to zero, promoting sparsity. This is particularly helpful in cluster expansion models, where many interactions may be negligible.

### B.2 Bayesian Regression

Bayesian regression treats model parameters as probability distributions rather than fixed values. It combines prior beliefs with observed data to produce a posterior distribution:

$$P(\beta | X, y) = \frac{P(y | X, \beta) \cdot P(\beta)}{P(y | X)}$$

This probabilistic framework allows for uncertainty quantification. A specific form used in this work is **Automatic Relevance Determination Regression (ARDR)**, which assigns each parameter a separate precision (inverse variance). Parameters deemed unimportant are automatically driven toward zero, yielding a sparse and interpretable model.

## Appendix C: Effective Cluster Interactions (ECIs)

The table below presents the Effective Cluster Interactions (ECIs) obtained from the cluster expansion model constructed for the  $\text{In}_x\text{Ga}_{1-x}\text{N}$  alloy system using the ICET package. Each entry corresponds to a symmetry-inequivalent cluster characterized by its order, radius, multiplicity, and sublattice configuration. The ECIs were extracted from the trained model and reflect the energetics of different atomic interactions across the alloy lattice. These values were used in Monte Carlo simulations to compute thermodynamic quantities such as mixing energy, free energy, and short-range order parameters.

The table provides the following information for each cluster:

- **Index:** Internal reference ID

## Appendix

- **Order:** Number of atoms in the cluster (e.g., 2 for pair, 3 for triplet)
- **Radius:** Average distance between cluster atoms
- **Multiplicity:** Number of symmetry-equivalent clusters
- **Orbit Index:** Symmetry orbit classification
- **Multicomponent Vector & Sublattices:** Species and site configuration
- **Parameter:** Raw fit coefficient
- **ECI:** Final effective interaction energy (in eV)

These parameters define the energy model used throughout the simulations presented in Chapter 2

index	order	radius	multiplicity	orbit_index	multicomponent_vect	sublattices	parameter	eci
0	0	0.0	1	-1	-	-	-0.045785	-0.045785
1	1	0.0	2	0	[0]	['A']	0.0	0.0
2	2	0.5	6	1	[0, 0]	['A', 'A']	0.038492	0.006415
3	2	0.5	6	2	[0, 0]	['A', 'A']	0.055882	0.009314
4	2	0.7071	6	3	[0, 0]	['A', 'A']	-0.018272	-0.003045
5	2	0.8165	2	4	[0, 0]	['A', 'A']	-0.004508	-0.002254
6	2	0.866	6	5	[0, 0]	['A', 'A']	-0.006692	-0.001115
7	2	0.866	12	6	[0, 0]	['A', 'A']	-0.020902	-0.001742
8	2	0.9574	12	7	[0, 0]	['A', 'A']	0.013844	0.001154
9	2	1.0	6	8	[0, 0]	['A', 'A']	0.0207	0.00345
10	2	1.118	12	9	[0, 0]	['A', 'A']	-0.018895	-0.001575
11	2	1.1902	12	10	[0, 0]	['A', 'A']	-0.008682	-0.000723
12	2	1.2247	6	11	[0, 0]	['A', 'A']	-0.003544	-0.000591
13	2	1.2583	6	12	[0, 0]	['A', 'A']	-0.007089	-0.001181
14	2	1.291	12	13	[0, 0]	['A', 'A']	0.0	0.0
15	2	1.3229	12	14	[0, 0]	['A', 'A']	-0.004852	-0.000404
16	2	1.3229	12	15	[0, 0]	['A', 'A']	-0.008161	-0.00068
17	2	1.354	6	16	[0, 0]	['A', 'A']	0.0	0.0
18	2	1.4434	12	17	[0, 0]	['A', 'A']	0.008896	0.000741
19	2	1.5	6	18	[0, 0]	['A', 'A']	0.008095	0.001349
20	2	1.5	6	19	[0, 0]	['A', 'A']	0.0	0.0
21	3	0.5774	2	20	[0, 0, 0]	['A', 'A', 'A']	0.004769	0.002384
22	3	0.5774	2	21	[0, 0, 0]	['A', 'A', 'A']	-0.005736	-0.002868
23	3	0.5773	12	22	[0, 0, 0]	['A', 'A', 'A']	0.001031	8.6e-05
24	3	0.654	24	23	[0, 0, 0]	['A', 'A', 'A']	0.008807	0.000367
25	3	0.6667	6	24	[0, 0, 0]	['A', 'A', 'A']	-0.007139	-0.00119
26	3	0.6875	6	25	[0, 0, 0]	['A', 'A', 'A']	0.0	0.0
27	3	0.6991	12	26	[0, 0, 0]	['A', 'A', 'A']	-0.006703	-0.000559
28	3	0.6991	24	27	[0, 0, 0]	['A', 'A', 'A']	-0.016517	-0.000688
29	3	0.7056	12	28	[0, 0, 0]	['A', 'A', 'A']	0.0	0.0
30	3	0.7979	12	29	[0, 0, 0]	['A', 'A', 'A']	0.0	0.0
31	3	0.7979	24	30	[0, 0, 0]	['A', 'A', 'A']	0.009747	0.000406
32	3	0.8278	24	31	[0, 0, 0]	['A', 'A', 'A']	0.0	0.0
33	3	0.8584	6	32	[0, 0, 0]	['A', 'A', 'A']	0.004433	0.000739
34	3	0.8654	12	33	[0, 0, 0]	['A', 'A', 'A']	0.005019	0.000418
35	3	0.8654	24	34	[0, 0, 0]	['A', 'A', 'A']	0.00355	0.000148
36	3	0.8792	24	35	[0, 0, 0]	['A', 'A', 'A']	0.00654	0.000272
37	3	0.9011	24	36	[0, 0, 0]	['A', 'A', 'A']	0.016561	0.00069
38	3	0.9168	12	37	[0, 0, 0]	['A', 'A', 'A']	0.000601	5e-05
39	3	0.9168	12	38	[0, 0, 0]	['A', 'A', 'A']	0.0	0.0
40	3	0.9168	24	39	[0, 0, 0]	['A', 'A', 'A']	0.0	0.0
41	3	0.925	12	40	[0, 0, 0]	['A', 'A', 'A']	-0.00225	-0.000187
42	3	0.9398	12	41	[0, 0, 0]	['A', 'A', 'A']	0.003795	0.000316
43	3	0.9398	12	42	[0, 0, 0]	['A', 'A', 'A']	-0.003323	-0.000277
44	3	0.9393	24	43	[0, 0, 0]	['A', 'A', 'A']	0.0	0.0
45	3	0.9744	24	44	[0, 0, 0]	['A', 'A', 'A']	-0.022165	-0.000924
46	3	0.9809	12	45	[0, 0, 0]	['A', 'A', 'A']	0.0	0.0
47	3	1.0	4	46	[0, 0, 0]	['A', 'A', 'A']	0.0	0.0
48	3	1.0351	12	47	[0, 0, 0]	['A', 'A', 'A']	0.0	0.0
49	3	1.0513	12	48	[0, 0, 0]	['A', 'A', 'A']	0.0	0.0
50	3	1.0704	24	49	[0, 0, 0]	['A', 'A', 'A']	0.0	0.0
51	3	1.1219	12	50	[0, 0, 0]	['A', 'A', 'A']	0.0	0.0
52	3	1.1547	2	51	[0, 0, 0]	['A', 'A', 'A']	0.0	0.0
53	3	1.1547	2	52	[0, 0, 0]	['A', 'A', 'A']	0.0	0.0
54	4	0.6124	4	53	[0, 0, 0, 0]	['A', 'A', 'A', 'A']	0.0	0.0
55	4	0.6545	12	54	[0, 0, 0, 0]	['A', 'A', 'A', 'A']	0.001987	0.000166
56	4	0.6545	12	55	[0, 0, 0, 0]	['A', 'A', 'A', 'A']	0.0	0.0
57	4	0.7071	6	56	[0, 0, 0, 0]	['A', 'A', 'A', 'A']	0.0	0.0



## References

- [1] J.A. Chan, J.Z. Liu, A. Zunger, *Phys. Rev. B* **82** (2010) 045112.
- [2] S. Lee, C. Freysoldt, J. Neugebauer, *Phys. Rev. B* **90** (2014) 245301.
- [3] K.K. Duff, A. Schleife, J.B. Varley, C. Gounaris, A. Janotti, C.G. Van de Walle, *Phys. Rev. B* **91** (2015) 085206.
- [4] R. Mohamad, A. Béère, J. Chen, P. Ruterana, *Phys. Status Solidi B* **254** (2017) 1700206.
- [5] A. Tabata, S. Fuke, T. Suzuki, H. Fujioka, *Appl. Phys. Lett.* **80** (2002) 769.
- [6] L. Lymperakis, M. Hocker, C. Giesen, S. Schulz, J. Neugebauer, *Phys. Rev. Materials* **2** (2018) 015401.
- [7] D. Doppalapudi, S.N. Basu, K.F. Ludwig, T.D. Moustakas, *J. Appl. Phys.* **84** (1998) 1389.
- [8] O. Ambacher, *J. Phys. D: Appl. Phys.* **31** (1998) 2653–2710.
- [9] M.A. Caro, *PhD thesis*, Aalto University, Finland (2014).
- [10] L. Lymperakis, *PhD thesis*, University of Paderborn, Germany (2005).

- [11] I. Mplvnh, *BSc thesis*, University of Crete (2024).
- [12] M. Gkanas, *Diploma thesis*, University of Crete, Greece (2023).
- [13] T.P. Bartel, P. Specht, J.C. Ho, C. Kisielowski, *Philos. Mag. A* **87** (2007) 1049–1061.
- [14] A. van de Walle, G. Ceder, *J. Phase Equilib.* **23** (2002) 348–359.
- [15] C. Stampfl, C.G. Van de Walle, *Phys. Rev. B* **65** (2002) 155212.
- [16] J. Buckeridge, D.O. Scanlon, A. Walsh, C.R.A. Catlow, *Comput. Phys. Commun.* **205** (2016) 106–121.
- [17] S.B. Zhang, J.E. Northrup, *Phys. Rev. Lett.* **67** (1991) 2339–2342.
- [18] A. Seko, A. Togo, F. Oba, I. Tanaka, *Phys. Rev. B* **73** (2006) 094116.
- [19] S.H. Wei, A. Zunger, *Appl. Phys. Lett.* **69** (1996) 2719–2721.
- [20] L. Bellaiche, D. Vanderbilt, *Phys. Rev. B* **61** (2000) 7877.
- [21] A.I. Duff, L. Lymperakis, J. Neugebauer, *Phys. Status Solidi B* **252** (2015) 855–865.
- [22] D. Sobieraj, J.S. Wróbel, T. Rygier, K.J. Kurzydłowski, O. El Atwani, A. Devaraj, E. Martinez Saez, D. Nguyen-Manh, *Phys. Chem. Chem. Phys.* **22** (2020) 23174–23191.
- [23] J.-E. Lee, A. Janotti, C.G. Van de Walle, *Phys. Rev. B* **90** (2014) 245301.

## References

---

- [24] Nakamura, S., & Fasol, G. (1997). *The blue laser diode*. Springer-Verlag.
- [25] M. Ångqvist, W.A. Muñoz, J.M. Rahm, E. Fransson, C. Durniak, P. Rozyczko, T.H. Rod, P. Erhart, *Adv. Theory Simul.* **2** (2019) 1900015.