UNIVERSITY OF OKLAHOMA GRADUATE COLLEGE

Global Optimization Strategies for Efficient Energy Landscape Exploration in Atomistic Simulations

A THESIS

Submitted to the Graduate Faculty

In partial fulfillment of the requirements for the Degree of

MASTER OF SCIENCE

By Michael Mark Nii Addo Quaynor

Norman, Oklahoma

2025

Global Optimization Strategies for Efficient Energy Landscape Exploration in Atomistic Simulations

A Thesis Approved for the

GALLOGLY COLLEGE OF ENGINEERING

By the Committee consisting of :

Dr. Kasun K. Gunasooriya Govini Thanthrige, Chair

Dr. Charles Nicholson

Dr. Dimitrios Diochnos

Table of Contents

Abstract

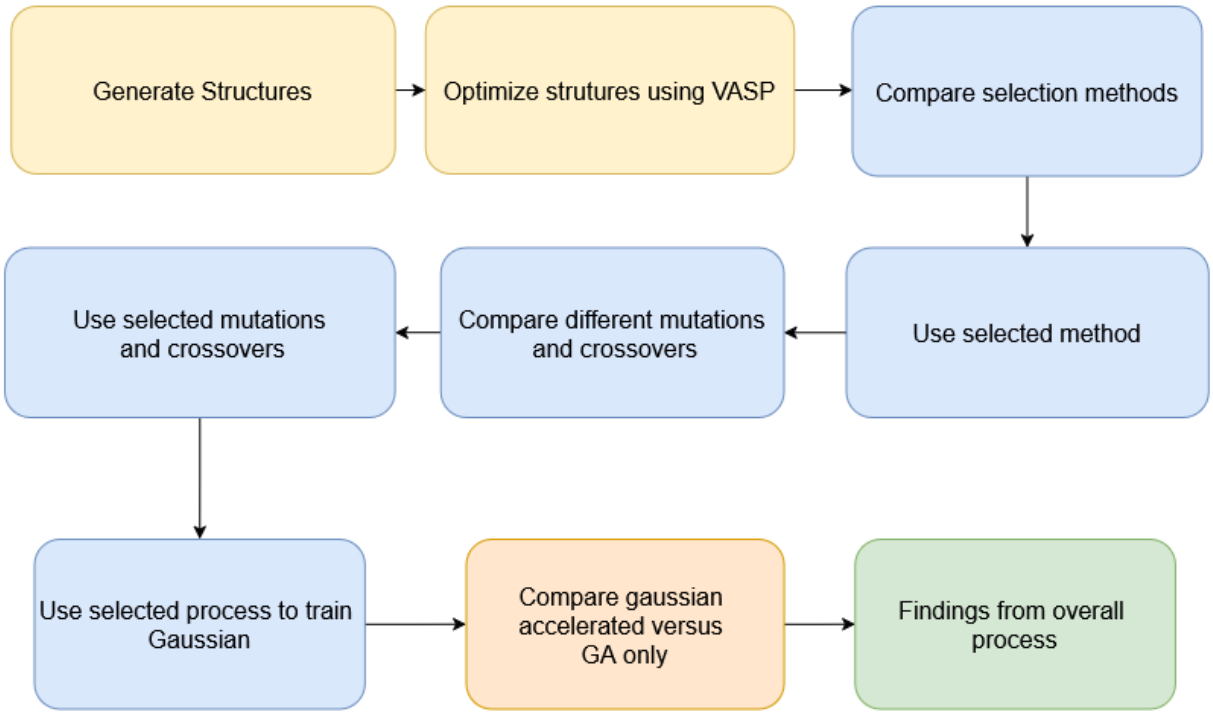


Figure 1: Summary of process.

Genetic algorithms (GAs) are widely used in materials science to explore complex potential energy surfaces and identify optimal atomic configurations. This study develops a comprehensive GA-based workflow to optimize nanoparticle structures, incorporating both traditional and machine learning-accelerated approaches. The process begins with the generation of candidate structures, followed by structural relaxation using the Vienna Ab initio Simulation Package (VASP).

We evaluate and compare five selection strategies: tournament, elitism, ranking, elitism with ranking, and elitism with tournament. These strategies are compared using hypothesis testing to assess statistically significant differences in optimization performance.

This analysis is conducted on two systems:

- An FCC(111) 3×3×4 Ag slab with a single atom of (Pd, Pt, Ru, or Rh),
- A Ni_6Pd_4 cluster, where the goal is to discover the atomic arrangement that minimizes total energy.

Various crossovers are benchmarked following the selection strategy comparison to further refine the evolutionary process. The best-performing GA configuration is then coupled with a Gaussian Process (GP) surrogate model, enabling accelerated prediction of structural stability and significantly reducing the number of expensive DFT evaluations.

Key findings from this study include:

- Ranking and tournament selection strategies effectively balance exploration and exploitation, yielding consistently low-energy configurations.
- Integrating a Gaussian Process model reduces the number of VASP evaluations needed by approximately three generations, achieving similar or better minima with minimal loss in accuracy for the cluster system.
- Simple Cut Splice and Half Uniform crossovers produced the most favorable distributions among the crossover operators tested.

These results highlight the critical importance of selecting appropriate evolutionary strategies and operators and demonstrate the potential of surrogate modeling to improve computational efficiency significantly in nanoparticle structure optimization. The combined GA-GP framework developed here offers a promising path forward for accelerating the discovery and design of advanced materials.

Code Availability:
The custom genetic algorithm operators (selection, crossover, mutation) and Gaussian Process surrogate modeling scripts developed as part of this work are available at:
https://github.com/gunasooriya-lab/cluster

Chapter 1: Introduction

1.1Background & Motivation

Genetic algorithms (GAs) are population-based metaheuristic optimization techniques inspired by natural selection and genetics [1]. These algorithms simulate evolutionary processes, where candidate solutions like individuals in a population undergo selection, crossover, and mutation over successive generations to create optimal solutions. Initially developed for optimization problems, GAs have been increasingly adopted in scientific and engineering fields due to their flexibility and robustness. In materials science, particularly in the design of catalysts, nanoparticles, and complex atomic systems, GAs are invaluable for navigating high-dimensional search spaces. Discovering optimal atomic arrangements often requires evaluating millions of possible configurations, many of which differ slightly in structure but significantly in stability or functionality [2].These problems are commonly classified as NP-hard, meaning the time required to solve them increases exponentially with system size, making brute-force or exhaustive search approaches computationally infeasible [2]. Traditional optimization techniques, such as gradient descent or linear programming, often fail in these contexts due to the discontinuous and highly non-linear nature of the potential energy surface (PES) that governs atomic interactions. In contrast, GAs are well-suited for such landscapes because they are stochastic[3] and capable of escaping local minima through recombination and mutation operations. This enables them to explore global energy minima in complex atomic systems efficiently. In catalyst design and nanoparticle optimization, GAs search for atomic configurations that minimize the system's total energy while optimizing properties such as catalytic activity, selectivity, thermal stability, and surface accessibility. By evolving a diverse population of candidate structures, GAs can identify previously unexplored configurations with desirable physical or chemical properties.

The effectiveness of GAs is enhanced by their integration with first-principles simulation tools such as the Vienna Ab initio Simulation Package (VASP), which is used to calculate total energies, forces, and other electronic structure properties of each candidate configuration. However, VASP calculations are computationally expensive, particularly for large systems or when many candidate structures must be evaluated. To address this bottleneck, recent approaches have coupled GAs with machine learning-based surrogate models, such as Gaussian Process Regression (GPR) or neural networks, which can approximate the energy landscape based on a subset of VASP-evaluated structures [3]. This hybrid strategy enables the algorithm to prioritize promising candidates while reducing the number of expensive simulations, thereby accelerating convergence without compromising accuracy.

This thesis is motivated by the need to develop and evaluate such hybrid GA-based frameworks for atomic structure optimization. Specifically, it focuses on improving selection and crossover strategies, benchmarking them statistically, and incorporating Gaussian Process models to accelerate search. The overall goal is to enhance the efficiency and effectiveness of computational discovery pipelines for novel catalyst and nanoparticle systems.

The complexity of these optimization problems stems from the potential energy surface (PES), a high-dimensional hypersurface mapping atomic configurations to their total energies. As noted by

Doye and Calvo[4] the PES of atomic clusters and nanoparticles becomes increasingly rugged with system size, exhibiting an exponential increase in local minima. This exponential scaling arises from the idea that a system of *N* atoms can be approximately divided into *M* independent or weakly interacting subsystems, each of size *N/M*. If each subsystem has $n_s(N/M)$ stable configurations, then the total number of stable configurations for the full system is given by[4]:

$$\mathrm{n_s(N)} = [\mathrm{n_s}((\frac{N}{M})^M]$$

Solving this recursive relation leads to the expression:

$$\mathrm{n_s(N)} = e^{\alpha N}$$

Here, $n_s(N)$ is the number of local minima for a system with *N* atoms, and α is a positive constant that characterizes the density of minima per atom. This result shows that the number of metastable configurations grows exponentially with system size. Computational studies, such as those on Lennard-Jones clusters[5] support this exponential behavior.

This motivates using Gaussian Process (GP) models as surrogate predictors of total energy. A GP is a non-parametric Bayesian model that assumes a smooth, continuous relationship between input features, such as structural descriptors, and outputs such as energy[6]. GP models are particularly well-suited for modeling PES landscapes for the reasons below:

- Smoothness Assumption: GPs assume that points close in structure space have similar energies, an assumption supported by the smooth nature of low-energy PES regions.[7]
- Uncertainty Quantification: GPs provide confidence intervals, enabling a balance between exploration (testing uncertain regions) and exploitation (testing low-energy regions)[8].
- Data Efficiency: GPs perform well with relatively small datasets, making them ideal for high-cost evaluation environments like DFT[8].

By training a GP model on the population of structures evaluated during GA runs, we can predict which unexplored structures are most likely to yield lower energies. This surrogate-assisted approach significantly reduces the number of expensive VASP calculations needed.

Several optimization methods have been explored for materials discovery, each with strengths and limitations:

- Random Search: Simple but inefficient due to lack of direction[2].
- Simulated Annealing: A probabilistic method that mimics the physical process of annealing by accepting worse solutions with a certain probability, allowing it to escape local minima. However, it requires careful tuning and long runtimes [9].
- Bayesian Optimization: Effective in low dimensions but scales poorly with increasing complexity.

Compared to these methods, GAs offer a flexible and robust framework. One of the most critical factors influencing their performance is the selection mechanism, the strategy used to choose

individuals for reproduction. The effectiveness of the selection strategy affects convergence speed, diversity retention, and ultimately the algorithm's ability to find the global minimum.

Common selection mechanisms include:

- Tournament Selection: Selects the best candidate from a random subset, offering a balance between diversity and selection pressure.
- Rank-Based Selection: Uses rank instead of raw fitness, reducing dominance by highly fit individuals and promoting more gradual convergence.
- Elitism: Ensures that the best solutions are carried forward to the next generation, protecting good candidates from being lost.
- Hybrid approaches, such as elitism with tournaments or elitism with ranking, attempt to combine rapid convergence with diversity preservation.

In this work, we systematically compare five GA selection strategies: tournament, elitism, ranking, elitism with tournament, and elitism with ranking using hypothesis testing to evaluate their effectiveness in guiding search convergence. The framework is applied to two systems:

1. An FCC(111) 3×3×4 Ag slab with a single atom of (Pd, Pt, Ru, or Rh),
2. A Ni_6Pd_4 nanoparticle cluster, where the objective is to identify the lowest-energy atomic arrangement.

All structures are relaxed and evaluated using VASP. After identifying the most effective GA configuration, we trained a Gaussian process model on the resulting dataset to accelerate further exploration. This hybrid GA-GP framework leverages the structure of the PES to reduce computational cost and increase efficiency in discovering optimal materials.

1.2 Research Objectives and Scope

The primary objective of this study is to develop and evaluate a genetic algorithm (GA)-based framework for optimizing atomic structures in catalytic and nanoparticle systems. To achieve this, we aim to:

1. Compare five GA selection strategies: tournament, elitism, ranking, elitism with tournament, and elitism with ranking.
2. Relax and evaluate all candidate structures using DFT calculations via VASP, ensuring accurate estimation of total energy.
3. Incorporate a Gaussian process surrogate model trained on GA-generated data to accelerate the optimization process, reduce computational cost, and enhance exploration of the potential energy surface (PES).

Through this framework, the study seeks to establish an efficient and scalable method for discovering optimal configurations in complex atomic systems, while also gaining insight into the influence of selection strategies on GA performance.

Chapter 2: Methodology

2.1 Initial Structure Generation

An initial population of 100 candidate structures is generated using a *Diverse Initial Population* class within our custom extension of the Atomic Simulation Environment (ASE) hosted on GitHub, which ensures structural diversity while adhering to known facets. For this system, a Face-Centered Cubic (FCC)(111) surface is used, modeled as a 3×3×4 slab, resulting in 36 atoms in total.

The composition of the slab is as follows: 32 Ag atoms and one atom each of Pt, Ru, Rh, and Pd, totaling 36 atoms. First, out of the 36 atomic positions, 32 positions are chosen to place the Ag atoms. This selection can be calculated using the following formula:

$$_{36}C_{32} = \frac{36!}{32! \times (35-32)!} = 58905$$

After fixing the Ag positions, there are four remaining sites to be filled by Pt, Ru, Rh, and Pd. Since these four atoms are all distinct, they can be arranged among the four available sites in 4! ways. Thus, the total number of unique configurations is:

$$_{36}C_{32} \times 4! = \frac{36!}{32! \times (35-32)!} \times 4! = 1413720 \text{ cobinations}$$

This large number of possible atomic arrangements highlights the combinatorial complexity of the system, demonstrating the necessity of using global optimization methods such as genetic algorithms (GAs) for efficient structure search.

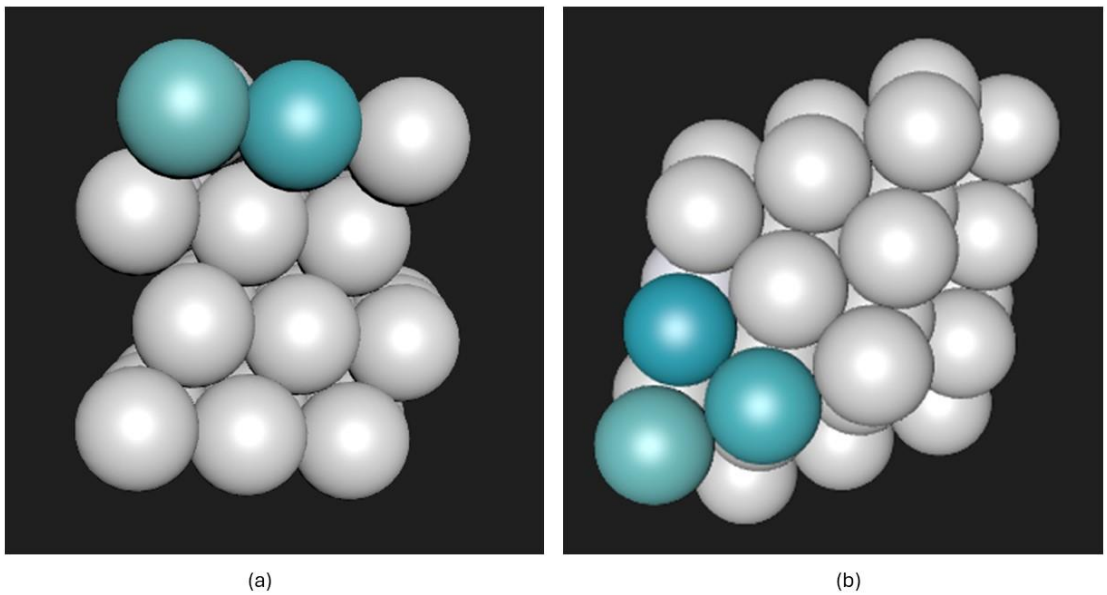


Figure 2: Side View (a) and Top View(b) of FCC(111) 3×3×4 slab.

The objective of the Ni_6Pd_4 nanoparticle cluster optimization is to identify the most stable atomic arrangement for a fixed composition of six nickel and four palladium atoms. The global minimum for this system is known to exhibit an octahedral geometry [9]. To generate an initial population of structurally diverse candidates, the configuration process begins by fixing one atom at the center of a 20×20×20 $Å^3$ simulation box. The remaining nine atoms are randomly distributed within the box, with their x, y, and z coordinates uniformly sampled within a limited range. This spatial constraint ensures that atoms are not placed too far apart initially, which helps prevent unphysical behavior such as fragmentation or excessive dispersion during the early stages of relaxation.

To enforce physical realism, the initial population is generated using the *Stable Initial Population* class, which applies a minimum interatomic distance constraint to avoid unrealistic overlaps and to preserve chemically meaningful bonding environments. The resulting population is compositionally constrained and structurally diverse, as a suitable starting point for evolutionary optimization.

The genetic algorithm is then used to apply selection, crossover, and mutation operators to evolve these structures toward lower-energy configurations.

Figure 3 illustrates a representative random structure generated using this method, while Figure 4 shows the known global minimum octahedral configuration used as a reference for evaluation.

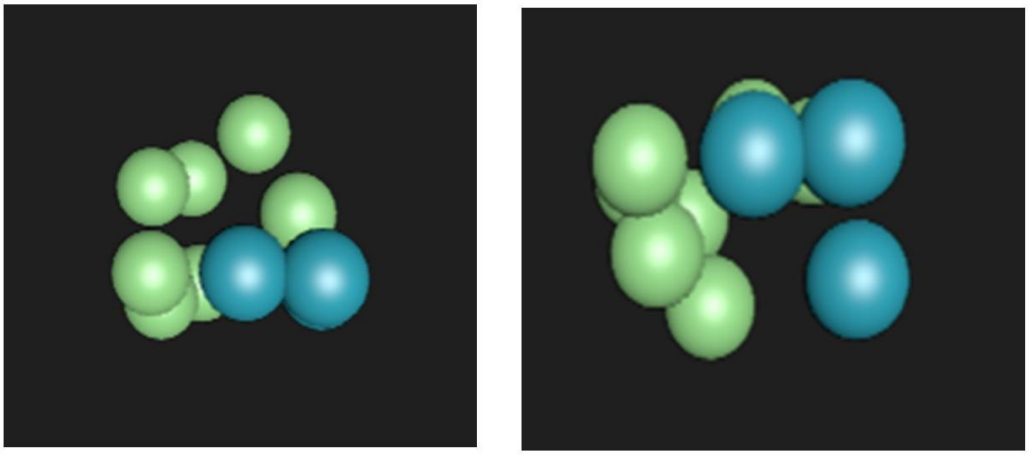
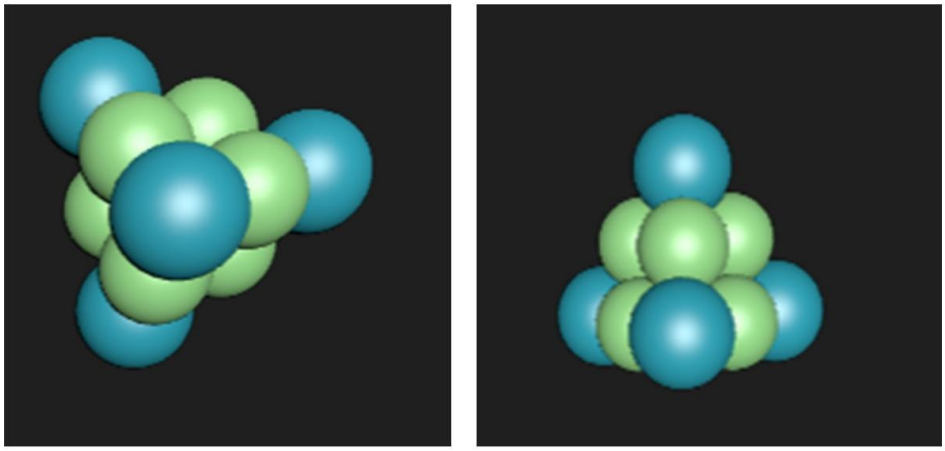


Figure 3: Side View (a) and Top View(b) of Ni_6Pd_4 sample generated cluster.



(a) (b)

Figure 4: Side view (a) and top view (b) of the globally optimized Ni_6Pd_4 cluster exhibiting the known octahedral geometry. This structure serves as the reference global minimum for comparison in the genetic algorithm optimization process.

2.2 Genetic Algorithm Implementation

This study's genetic algorithm (GA) framework is designed to explore complex atomic configuration spaces and identify low-energy structures for both slab and nanoparticle systems. The implementation features a population-based optimization loop, multiple custom crossover and mutation operators, and direct integration with VASP for high-fidelity energy evaluation.

The fitness of each candidate structure is determined by its total energy, computed through density functional theory (DFT) using VASP. This involves numerically solving the Kohn–Sham formulation of the Schrödinger equation[10], enabling accurate estimation of electronic structure and total energy. This physics-based fitness function ensures that a reliable and consistent measure of structural stability guides the GA.

2.2.1 Crossover Operators

Crossover operators are fundamental to the evolutionary search, enabling the combination of structural features from parent configurations to generate new candidate structures. Several crossover strategies are implemented to preserve diversity and enforce physical validity. The crossovers are the following:

- Three Parents Crossover: This randomly selects an atom at each index from one of three parent structures. This promotes higher diversity and reduces the likelihood of stagnation.

- Matrix Crossover: This uses a probability to decide whether each atom is inherited from the first or second parent. This probabilistic approach allows control over the level of mixing.
- Half Uniform Crossover: Splits the atomic list and combines the first half from one parent and the second half from the other, preserving the ordering.
- Multi Point Crossover: Introduces multiple crossover points to alternate atomic segments between two parents, creating novel structures.
- Simple Cut-Splice Crossover: Adapted from the Alloy Catalysis Automated Toolkit (ACAT) [10], this crossover operator slices both parent structures along a randomly oriented 3D plane and exchanges atoms on one side of the plane. The method is spatially aware, enabling constraints based on atom types or specified atom subsets, making it particularly effective for generating offspring in nanoparticle and surface model optimizations.

All crossover classes include a keep composition check to ensure that the elemental ratios in offspring match those of the parent configurations. This is essential for preserving physical and chemical consistency.

2.2.2 Mutation Operators

Mutation operators are applied to introduce local neighborhood changes in candidate structures, encouraging exploration and preventing premature convergence. Each mutation inherits from a shared Mutation base class that provides utilities for calculating interatomic distances and preserving composition. The mutations are the following:

- Translational Mutation: This randomly displaces the entire structure by a vector of defined magnitude, aiding in exploring orientation space.
- Rotational Mutation: This rotates the structure about a random axis, enabling sampling of configurations inaccessible through translation alone.
- Shell Swap Mutation: This exchanges atoms from the innermost and outermost radial shells if they belong to different elements. This is effective for simulating surface core–shell rearrangements.
- Local Cluster Perturbation: This slightly displaces atoms within a radius around a randomly selected center atom, mimicking localized distortions.
- Element Substitution Mutation: This changes the type of a randomly selected atom to another from a predefined list, supporting compositional variation such as doping .
- Bond Perturbation: This identifies atom pairs closer than a threshold and changes the bond angle between the atoms.

Each mutation method is designed to produce physically meaningful variations while preserving structural feasibility.

2.2.3 Workflow Summary

The genetic algorithm begins with the generation of 100 initial structures, which are then relaxed using VASP to obtain accurate energy values. A selection strategy is applied to identify the most promising candidates for reproduction. Using the crossover and mutation operators described above, a selected crossover and mutation are selected to generate 50 new offspring structures, which are then relaxed using VASP. This loop selection, variation, and evaluation is repeated for five generations. After completing the optimization, the performance of each selection strategy is assessed. This GA framework provides a robust and extensible platform for structural optimization in materials design, balancing exploration of the potential energy surface with keeping physically possible structures after manipulation.

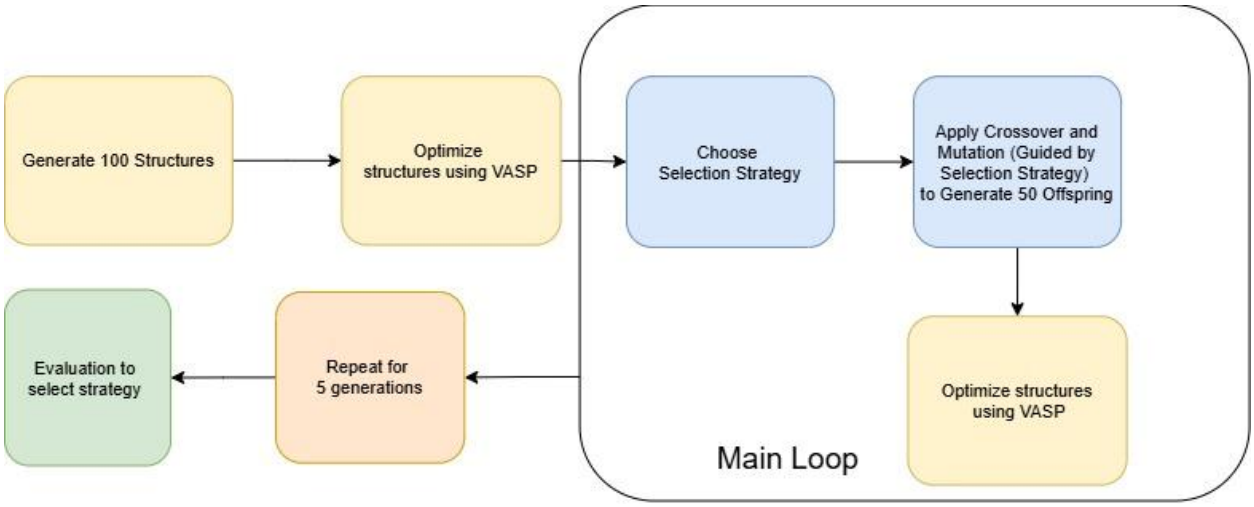


Figure 5:Summary of Genetic Algorithm process.

2.3 Gaussian Algorithm Implementation

To accelerate the discovery of low-energy atomic configurations, a GP model is employed to predict the total energy of candidate structures based on their atomic arrangements and associated physical properties. A GP is a machine learning process where one assumes that there is a formula that connects two points. This is estimated using a covariance kernel. Specifically for this problem, we assume that energy landscapes consist of known points and try to predict the energy of the unknown points using the energy of the known points, which are the slabs and clusters. This relationship is modeled through a covariance kernel, which defines the similarity between any two points in feature space. This surrogate model reduces the number of expensive VASP calculations while leveraging the structured nature of the potential energy surface (PES).

The dataset is organized into a Pandas DataFrame, where each row corresponds to a unique atomic structure and each column represents a numerical descriptor. The target variable for regression is the total DFT-calculated energy of each structure. To prepare the data for modeling, features are standardized using StandardScaler [11], and an 80/20 train-test split is applied with 5-fold cross-validation. Structural descriptors for the training set are extracted by processing VASP output files.

From the CONTCAR files, pairwise atomic distances are computed, normalized, and used as geometric features. OUTCAR files provide atomic forces, which serve as physically meaningful indicators of how far a structure is from its relaxed state, offering insight into the stability of the configuration. In addition to these features, Voronoi volumes and coordination numbers are computed using Pymatgen's local environment analysis tools [11], capturing local atomic packing and bonding environments. These combined descriptors enable the GP regression model to learn from each configuration's structural geometry and relaxation-relevant properties.

The GP model is implemented using GaussianProcessRegressor from scikit-learn[12]. Initially, a basic GP model is trained using a kernel composed of a constant term multiplied by an RBF kernel. This model provides both mean predictions and uncertainty estimates (standard deviations), allowing for an evaluation of how confident the model is in its predictions. The kernel takes the form:

$$\mathrm{K}(\mathrm{x},\mathrm{x}') = \mathrm{C}e^{\left(-\frac{|x-x'|^2}{2l^2}\right)}$$

Where:

- C: A constant kernel or scaling factor that determines the overall magnitude of the covariance. It controls the scale of the function, adjusting kernel values to align with data variability.
- e : Ensuring that similarity decreases exponentially with increasing distance. This makes the kernel suitable for continuous functions.
- $|x - x'|^2$: This is the distance between two data points. It measures their separation in feature space.[12]

In order to account for non-Gaussian datasets, a WhiteKernel [13] is added to add the non-Gaussian behavior to the model:

$$\mathrm{K}(\mathrm{x},\mathrm{x}') = \mathrm{C}e^{\left(-\frac{|x-x'|^2}{2l^2}\right)} + \mathrm{C}\sigma^2$$

Here, the equation includes a noise component ($\mathrm{C}\sigma^2$), which is the variance of the data to account for non-Gaussian behavior or modeling inaccuracies, such as variability in DFT outputs or imperfect features. This addition helps prevent overfitting and reflects uncertainty in energy predictions. Additionally, a grid search is performed to optimize the kernel configuration, varying both the length scale and the noise level of the WhiteKernel. Each model created in the grid search is evaluated by training a GP model and computing the Mean Squared Error (MSE) on the test set:

The components of the MSE are :

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y - y')^2$$

- n: This is the number of data points in the dataset.
- y: This is the true energy value of the target of the structure calculated by DFT.
- y': This is the predicted energy value of the target variable for a structure from the Gaussian Process model.
- $y - y'$: This is the difference between the actual and predicted energy values for the data points, representing the prediction error.

The model with the lowest MSE is the best-fit surrogate model. Once trained, the GP model is applied to the full dataset to predict the energies of all candidate structures. The top 10 lowest-energy candidates are selected for further investigation based on the predicted mean energy values. The corresponding file paths of these candidates are retrieved and passed on for additional DFT validation and analysis, ensuring that the surrogate model's predictions align with high-accuracy energy evaluations. The overall process is summarized in Figure 6 below.

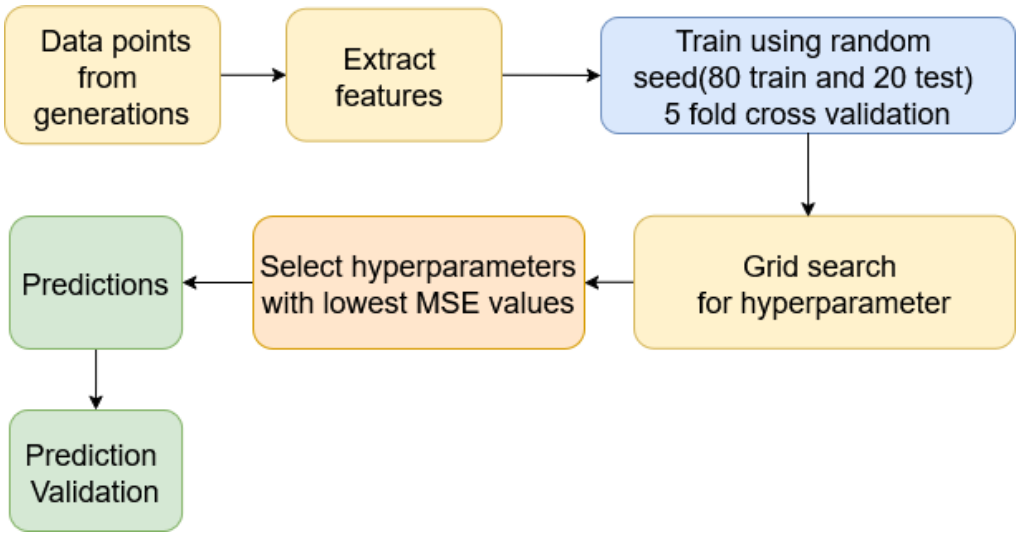


Figure 6:Summary of Gaussian process.

2.4 Experimental Setup

All structural optimizations and energy evaluations in this study were performed using VASP interfaced through the Atomic Simulation Environment (ASE).The workflow, including structure generation, geometry relaxation, and data conversion, was executed manually by submitting job scripts (sh files) to OU Supercomputing Center for Education & Research(OSCER).

Each structure, whether generated as part of an initial population or through crossover, was individually processed using a dedicated Python script that initializes the atomic system, assigns a VASP calculator, runs the energy evaluation, and stores the results. After each VASP job was completed, the results were manually saved to CSV files, which contained the path and energy of

parents and the generation number to ensure consistency in data management and later use in machine learning workflows.

Cluster Relaxation Procedure

For the Ni_6Pd_4 bimetallic clusters, the input atomic configuration was read from an init.traj file, which contains information on atomic positions and species. The VASP calculator was configured using ASE with the following key parameters: a plane-wave cutoff energy (encut) of 400 eV, the functional with PBE exchange-correlation. Ionic relaxation was conducted using the quasi-Newton algorithm (ibrion = 2), with a maximum of 100 ionic steps (nsw = 100). A minimal k-point mesh of (1, 1, 1) was used due to the isolated nature of the cluster[14].

After assigning the calculator to the atoms object, the total energy was computed and written to the energy.out file and slurm.out file after being optimized in OSCER. The relaxed structure was saved in a trajectory file (final.traj). These steps were repeated manually for each cluster in the population, ensuring consistent relaxation across generations.

2.4.2 Slab Relaxation Procedure

For the FCC(111) slabs, a vacuum of 5 was applied along the z-direction to prevent periodic image interactions, while periodic boundary conditions were enforced in the x and y directions.

The VASP calculator was set up similarly to the cluster case, with a 400 eV energy cutoff and PBE functional, but with a denser k-point mesh of (3, 3, 1) to account for the periodic surface geometry. The same relaxation scheme and convergence criteria were used, ensuring consistency between both system types. As with the cluster workflow, the energy of each slab was computed and saved, and the relaxed structure was stored as final.traj.

2.4.3 Justification of Slab-Only Benchmarking

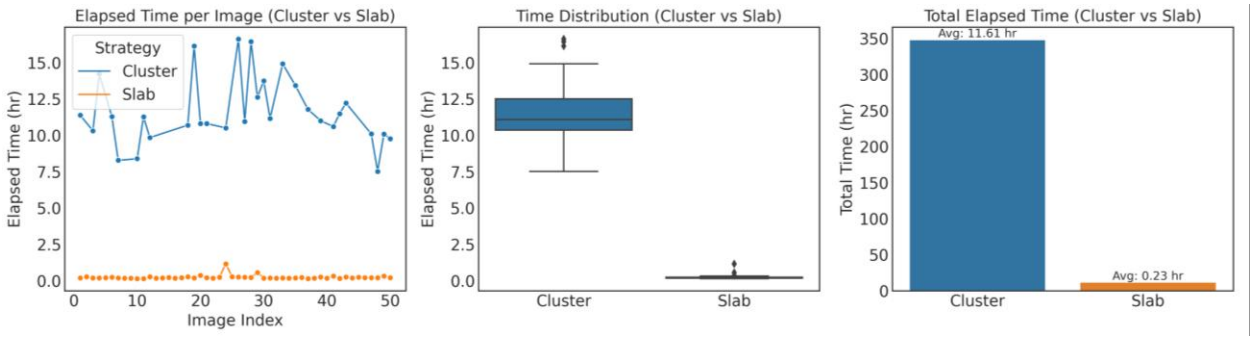


Figure 7: Slab versus cluster comparison.

Due to the substantial computational cost associated with structural relaxation using VASP, the full five-generation comparison across all selection strategies and crossover methods was conducted exclusively for the FCC(111) slab system. In contrast, the cluster system was evaluated only using the Gaussian Process (GP)-accelerated approach and the best crossover and selection strategy. This decision is justified by the runtime comparison presented in Figure 7, which includes three subplots: relaxation time per structure, distribution of relaxation times, and total computation time for each system.

The relaxation time per structure is shown in the left panel for 50 configurations. Blue points represent the cluster system, where each relaxation consistently requires between 8 and 16 hours, with an average of approximately 11.6 hours per structure. Conversely, the slab structures (orange) exhibit dramatically faster relaxation times, averaging just 0.23 hours per structure. This significant disparity highlights the efficiency of slab-based simulations.

The middle panel presents a boxplot of the relaxation time distributions. The cluster system displays a much broader range and higher variance in convergence times than the slab system. Numerous outliers are observed in the cluster data, indicating inconsistencies in relaxation behavior, likely stemming from differences in atomic packing, elemental composition, and initial configuration randomness.

The right panel summarizes the total computational effort for relaxing 50 structures in each system. The cumulative runtime for the cluster system approaches 350 hours, while the slab system completes all relaxations in approximately 12 hours. This stark contrast underscores the impracticality of conducting full-scale cluster experiments across multiple genetic algorithm strategies within feasible time constraints.

As such, the FCC(111) slab system was selected as the benchmark platform for comprehensively comparing selection strategies. The slab's periodic structure and well-defined lattice provide more constrained initial configurations, facilitating faster and more reliable convergence in VASP. In contrast, clusters lack this regularity, resulting in a more complex potential energy surface with more local minima. Consequently, relaxation from random initial configurations is significantly more time-consuming. These observations emphasize the computational burden of cluster-based simulations and the critical importance of system choice when designing and evaluating global optimization workflows.

2.5 Evaluation of Metrics and Statistical Analysis

Several performance metrics were used throughout the optimization process to evaluate and compare the effectiveness of different genetic algorithm selection strategies. These metrics were selected to assess the quality of solutions and each method's efficiency and exploration versus exploitation capabilities.

Convergence speed was measured as the number of generations required to reach a near-optimal or minimal energy solution. This metric reflects how quickly a given selection strategy can guide

the population toward favorable regions of the potential energy surface. A lower number of generations to convergence indicates faster and more efficient optimization.

The best fitness per generation, the lowest total energy of a candidate in that generation, was tracked to monitor the quality of the best solutions over time. This provides insight into whether the strategy consistently finds better structures or stagnates due to premature convergence.

To assess the diversity of the population, the standard deviation of fitness values (energies) was computed for each generation. A higher standard deviation suggests greater diversity, which is crucial for exploring and avoiding local minima. Conversely, a sharp decline in diversity may indicate convergence or loss of genetic variability.

Finally, computational efficiency was evaluated by recording the time taken to complete all generations under each selection strategy. This includes the time spent on VASP evaluations. Comparing time across strategies helps identify trade-offs between accuracy and computational cost, especially when scaling the method to larger systems.

These metrics were combined with statistical hypothesis testing to determine whether the observed differences between selection and crossover strategies were statistically significant. This approach ensures that conclusions about performance are supported by quantitative evidence.

Chapter 3: Selection Strategy Analysis

3.1 Overview of Strategies

This study systematically evaluates five selection strategies within a genetic algorithm (GA) framework to determine their effectiveness in discovering low-energy atomic configurations. The strategies tested are:

1. Tournament Selection
2. Elitism with Ranking Selection
3. Ranking Selection
4. Elitism Selection
5. Elitism with Tournament Selection

The evaluation begins with a common initial population of 100 relaxed structures, each optimized using VASP to determine its energy. From this shared starting point, each strategy is independently applied to produce 50 new offspring per generation using the Simple Cut Splice Crossover operator, ensuring that comparisons are based solely on the selection mechanism rather than differences in crossover or mutation. This process is selected, crossover, structure generation, and energy evaluations are repeated over five generations. All resulting structures, associated energy values, and metadata are stored for downstream analysis. Performance across the five strategies is then assessed using the evaluation metrics described in Section 2.5, including convergence speed and best fitness per generation.

The strategy demonstrating the best overall performance across these metrics is chosen for integration into the final GP-assisted optimization workflow. A schematic of the full workflow is presented in Figure 8 below.

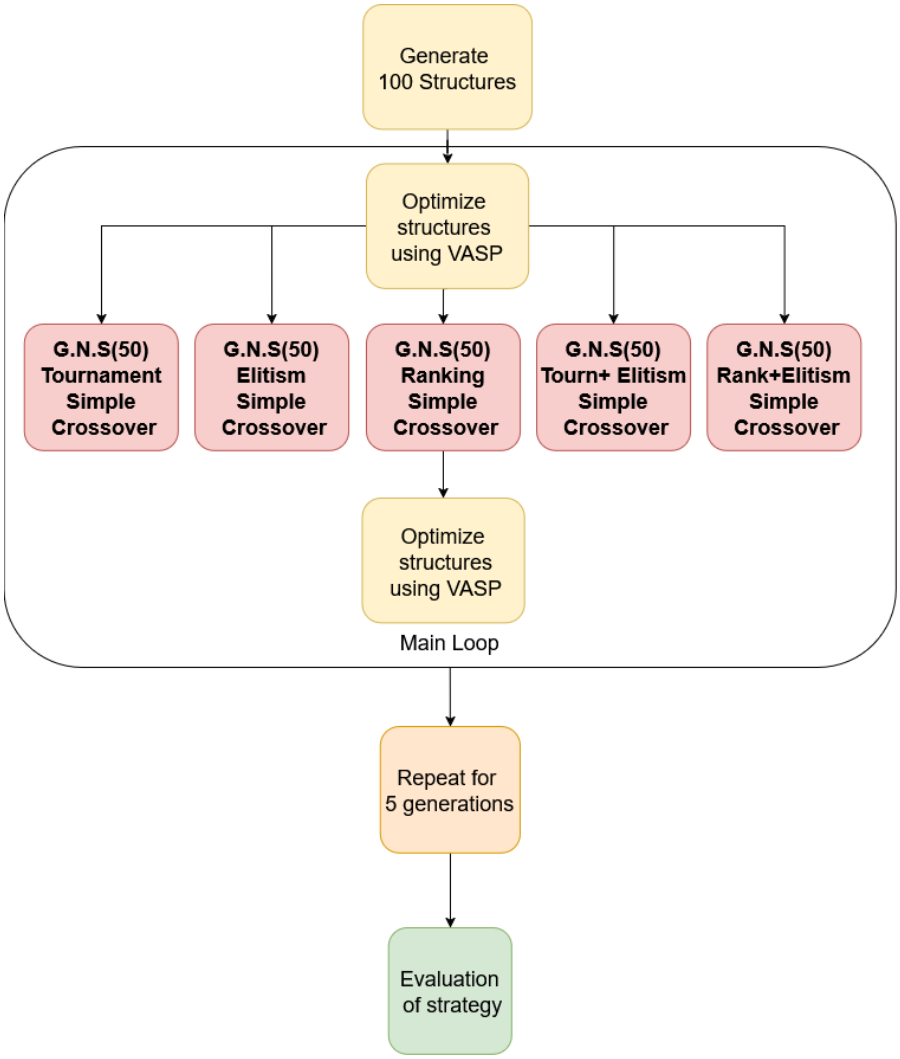


Figure 8: Overview of the genetic algorithm selection process. "G.N.S (50)" refers to Generate New Structures (50), representing the production of 50 offspring through crossover and from selected parent structures in each generation. The flow diagram summarizes the selection, reproduction, and evaluation steps used to evolve low-energy atomic configurations.

3.1.1 Tournament Selection

The Tournament Selection strategy selects parents by randomly sampling a group of k=5 individuals from the lowest 20 and choosing the one with the lowest energy. This process is repeated to obtain two distinct parents for each offspring. In every generation, 50 offspring are created using tournament-selected parents. This method balances exploration and exploitation, helping avoid premature convergence by maintaining genetic diversity while favoring better-performing candidates[15].

3.1.2 Ranking Selection

In Ranking Selection, individuals are ranked by energy, and selection probabilities are assigned based on these ranks rather than raw energy values. The top 20 individuals form the parent pool, and 50 offspring are generated using rank-weighted sampling and crossover. This strategy reduces

selection pressure, which is how strongly the algorithm favors top-ranked individuals [16]. A higher value of k increases the likelihood of selecting higher-ranked (lower-energy) individuals, accelerating convergence but risking premature loss of diversity, giving even lower-ranked individuals a nonzero chance of being chosen, thereby maintaining greater diversity across generations.

The selection probability P_i for individuals is given by the following formula:

$$P_i = \frac{\frac{1}{r_i}}{\sum_{i=1}^{n} \frac{1}{r_i}}$$

where:

- r_i is the rank of individual i (i.e., 1 for best, 2 for second-best, etc.),
- N is the total number of individuals in the parent pool,
- The denominator normalizes the probabilities so that $P_i = 1$

3.1.3 Elitism Selection

Elitism is a method where the top 20 lowest-energy structures are passed directly into the next generation. The remaining 30 offspring are generated using parents randomly selected from this elite group. This strategy accelerates convergence by preserving high-quality solutions but may reduce diversity over time if not paired with exploration components. Its strength lies in retaining progress made in previous generations.

3.1.4 Elitism with Ranking Selection

This hybrid strategy merges the benefits of elitism and ranking selection. The top 20 structures based on energy are retained unaltered, ensuring the best solutions persist. The remaining 30 offspring are generated using rank selection from the elite pool. Individuals are assigned selection probabilities based on their rank, with better individuals having higher chances of selection. This approach maintains diversity while promoting steady convergence and effectively avoids local minima in energy landscapes.

3.1.5 Elitism with Tournament Selection

This combined strategy uses elitism to carry over the top 20 structures, while tournament selection (with k=5) is applied within the elite pool to select parents for producing 30 additional offspring. This method blends the exploitative power of elitism with the exploration of tournament selection, helping to generate high-quality yet diverse candidates around the best-known solutions.

.

3.2 Results and analysis

3.2.1 Energy Distribution and Variability Analysis

The distribution and variability of offspring energies across different GA selection strategies are analyzed using three visualizations: boxplots, kernel density estimation (KDE) curves, and a standard deviation summary. These plots highlight differences in energy spread, central tendency, and consistency among the strategies.

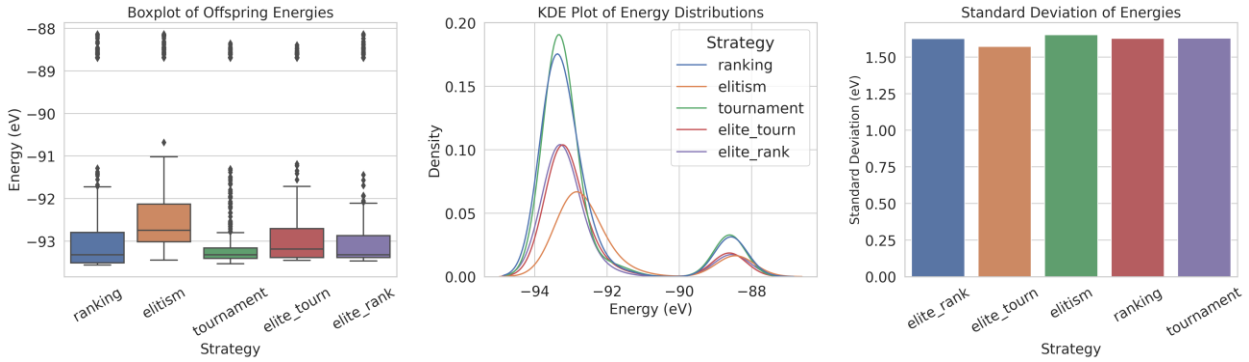


Figure 9: Visualization of offspring energy distributions across all genetic algorithm selection strategies. The figure includes box plots, kernel density estimation (KDE) curves, and a standard deviation summary to illustrate the variability, central tendency, and spread of energies produced by each strategy.

The boxplot provides a visual summary of the energy distribution across all offspring generated by each strategy. The ranking strategy yields the lowest median energy and a relatively compact interquartile range, which shows minor deviations. In contrast, the elitism strategy shows a higher median and wider range, suggesting higher variability in offspring quality. Both tournament and elitism with tournament exhibit competitive medians but also display more outliers, indicating occasional poor-performing structures.

The KDE plot shows a smooth estimate of the energy probability density for each strategy. The ranking strategy exhibits a clear peak at lower energy values with a narrow spread, highlighting its consistent performance. The elitism curve is broader and skewed toward higher energies, suggesting frequent generation of suboptimal structures. Similar to the ranking strategy, tournament exhibits a peak at lower energy values with a narrow spread.

The bar chart quantifies the standard deviation of energy for each strategy. The elitism strategy exhibits the lowest standard deviation, which may seem favorable at first, but is misleading when considering its higher average energy, indicating early convergence to suboptimal solutions. In contrast, the ranking strategy shows a slightly higher spread but significantly better energy performance overall. This reflects a good balance between exploration and exploitation.

3.2.2 Comparative Analysis: Ranking vs. Tournament Selection

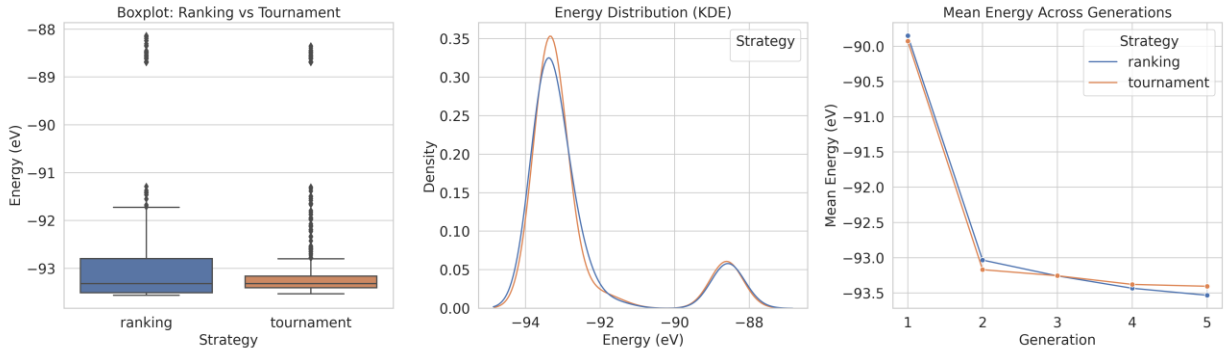


Figure 10: Comparison between Ranking and Tournament

Based on the observations above a comparison was made between the top two performing selection strategies: ranking and tournament. This analysis includes visual and statistical assessments of energy distributions for all offspring generated across five generations. The ranking and tournament strategies exhibit the largest and most consistent improvements in offspring energy over generations, confirming their effectiveness in generating energetically favorable configurations.

3.2.3 Paired Statistical Test: Ranking vs. Tournament Selection

In order to assess the relative performance of ranking and tournament selection, a paired t-test was conducted under the null hypothesis that the mean energies of the two strategies are equal, with an alpha of 0.05. This method involves comparing the energies of structures, ensuring a direct one-to-one pairing for fair evaluation.

The t-test results resulted in a p-value of 0.564 and a t-statistic of 0.577, indicating no statistically significant difference in energy values between the ranking and tournament selection strategies. While the ranking strategy exhibited better convergence trends overall, this test confirms that the two selection strategies perform similarly.

3.3 Conclusion

While the ranking strategy remains favorable for its smoother convergence and tighter control over diversity, tournament selection yields statistically comparable results in terms of energy minimization. This validates it as a competitive alternative, especially when ranking may be less interpretable.

3.4 Summary

The table below summarizes the strengths and search behavior associated with each genetic algorithm selection strategy based on observed energy trends, diversity, and convergence profiles.

Strategy	Search Behavior	Justification
Tournament	Exploitation	Achieves consistently low mean energy and low standard deviation; favors strong solutions.
Ranking	Exploration	Found the best individuals; wider energy range with higher standard deviation supports diversity.
Elitism	Strong Exploitation	Preserves top performers across generations but often at the cost of population diversity.
Elitism + Tournament	Balanced	Maintains low energy while preserving some exploration through tournament sampling.

Table 11: Summary of GA Selection Strategy Findings.

Chapter 4: Crossover Strategy Evaluation

4.1 Overview of Strategies

Following the evaluation of selection strategies, the next phase of the study focuses on comparing the performance of different crossover operators within the GA framework. As illustrated in Figure 11, the experimental setup begins with generating an initial population of 100 structures optimized using VASP to establish a baseline energy landscape. The population is divided into separate evolutionary paths from this starting point, each corresponding to a different crossover operator. The crossover methods tested include Simple Cut Splice, Three Parent, Matrix, Half Uniform, and MultiPoint crossovers. Each crossover operator is paired with the tournament selection strategy to ensure consistent parent selection across experiments. In each crossover test, 50 new offspring are generated per generation using the assigned crossover operator, and all resulting structures are again optimized using VASP. This process is repeated for five generations, allowing each crossover strategy to evolve its population toward lower-energy configurations independently. At the end of the five generations, each crossover method's energy performance, convergence trends, and diversity of offspring produced are evaluated and compared. This comparison highlights the impact of different recombination mechanisms on search efficiency and solution quality in the global optimization process.

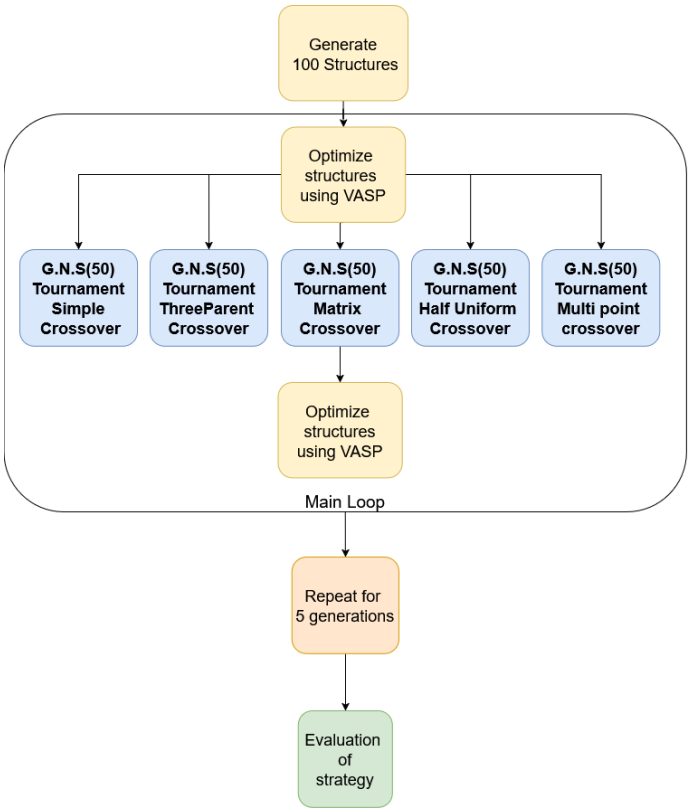


Figure 11: Overview of the genetic algorithm crossover process. "G.N.S (50)" refers to Generate New Structures (50), representing the production of 50 offspring through crossover from selected parent structures in each generation. The flow diagram summarizes the selection, reproduction, and evaluation steps used to evolve low-energy atomic configurations.

4.2 Results and Analysis

The impact of different crossover operators on genetic algorithm performance was evaluated based on the energy distributions of the offspring they produced. Figure 12 shows the results through boxplots, kernel density estimation (KDE) plots, and a comparison of standard deviations of offspring energies across the five crossover strategies.

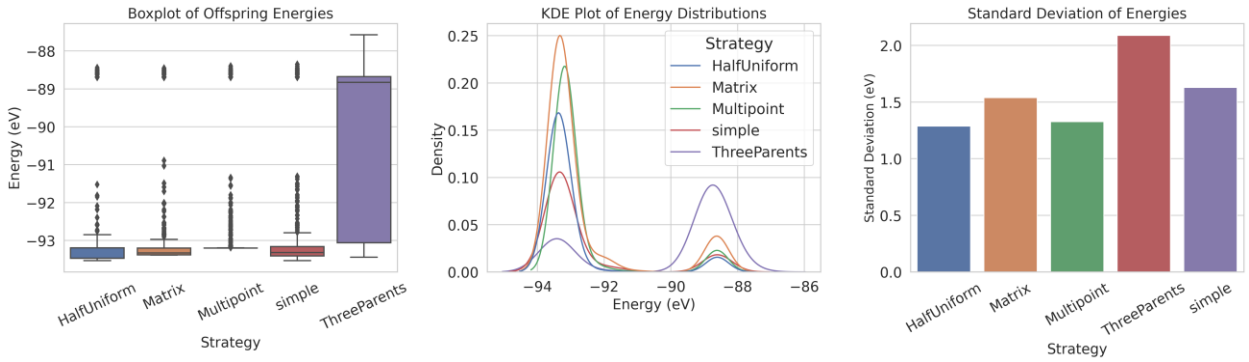


Figure 12: Visualization of offspring energy distributions across all genetic algorithm crossover strategies. The figure includes box plots, kernel density estimation (KDE) curves, and a standard deviation summary to illustrate the variability, central tendency, and spread of energies produced by each strategy.

The boxplot shows the distribution of offspring energies for each crossover method. Simple Cut Splice, Half Uniform, and Matrix crossover strategies produced offspring with consistently lower median energies and narrower interquartile ranges compared to the others. In contrast, the Three Parent crossover shows a much higher median energy and a wider spread, indicating poorer convergence and higher variability in offspring quality. The KDE plot provides further insight into the shape and spread of the energy distributions. The Simple Cut Splice and Half Uniform crossovers exhibit narrow, sharp peaks at lower energies, suggesting that these methods consistently produce low-energy offspring. Matrix and MultiPoint crossovers also show relatively favorable distributions but slightly broader peaks. The Three Parent crossover exhibits a wider, flatter distribution shifted toward higher energies, reinforcing its lower effectiveness for this system. The standard deviation bar plot quantifies the variability of energies each crossover operator produces. Three Parent crossovers have the highest standard deviation, consistent with its poor energy convergence and broad distribution. Matrix and Simple Cut Splice crossovers exhibit lower standard deviations, indicating better control over offspring quality and more stable convergence behavior.

4.2.1 Comparative Analysis: Simple Cut Splice vs. Half Uniform Crossover

Given that Simple Cut Splice and Half Uniform crossover operators exhibited the best overall performance in earlier comparisons, a direct analysis was conducted to compare their effectiveness more precisely over five generations. Figure 13 presents the results through box plots, mean energy trends, and lowest energy convergence curves.

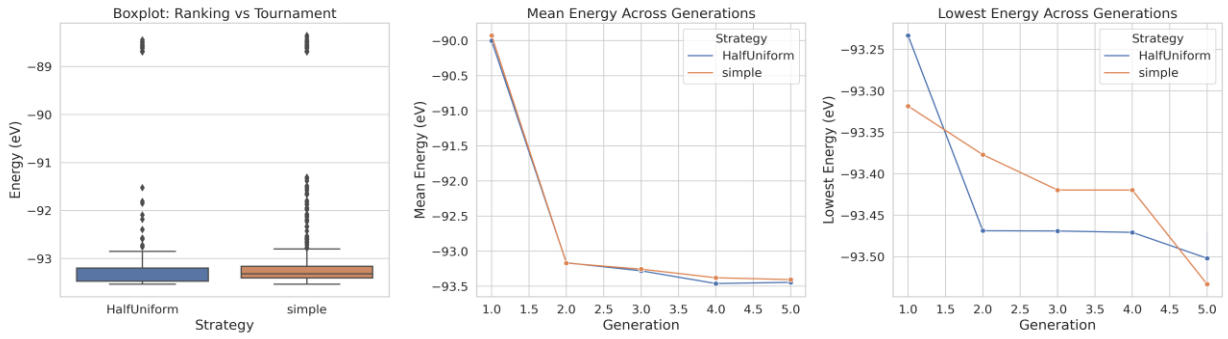


Figure 13: Comparison of offspring energy performance between Half Uniform and Simple Cut Splice crossover strategies under different GA selection mechanisms.

The boxplot comparing offspring energies between the two crossover strategies shows that Simple Cut Splice achieves slightly lower median energies and a more compact interquartile range than Half Uniform. However, both methods consistently produce low-energy structures, and their overall spread remains relatively tight compared to the rest of the crossover strategies.

The mean energy trend plot highlights the evolutionary dynamics over generations. Simple Cut Splice exhibits a steeper decline in average energy from Generation 1 to Generation 2 and maintains a marginally lower mean energy across subsequent generations. Half Uniform also shows effective energy reduction but plateaus slightly earlier, indicating that Simple Cut Splice offers better continuous improvement. Tracking the lowest energy per generation reveals that Simple Cut Splice consistently discovers lower-energy configurations compared to Half Uniform, especially after the third generation. By the fifth generation, Simple Cut Splice achieves the lowest absolute energy among all structures produced, confirming its superior ability to refine offspring toward global minima.

4.2.2 Paired Statistical Test: Half Uniform vs. Simple Cut Splice

To assess the relative performance of the Half Uniform and Simple Cut Splice crossover strategies, a paired *t*-test was conducted under the null hypothesis that the mean offspring energies of the two methods are equal, using an alpha of 0.05. This statistical test compares the energies of structures matched by generation and strategy, ensuring a fair one-to-one pairing.

The *t*-test yielded a *t*-statistic of -3.0342 and a *p*-value of 0.000245, indicating a statistically significant difference between the two crossover strategies. The lowest energy observed was −93.530 eV for Half Uniform and −93.5331 eV for Simple Cut Splice.

4.3 Conclusion

Simple Cut Splice, and Half Uniform crossover operators perform well compared to other tested methods. However, Simple Cut Splice crossover demonstrates a slight but consistent advantage, achieving lower mean energies, tighter offspring distributions, and the best minimum energies over multiple generations. Based on these findings, Simple Cut Splice is selected as the preferred crossover operator for the final genetic algorithm framework.

4.4 Summary

Crossover	Search Behavior	Justification
Simple Cut Splice	Exploitation and Fine-Tuning	Achieves the lowest mean and minimum energies; maintains a tight energy distribution across generations.
Half Uniform	Balanced Exploration and Exploitation	It produces low-energy offspring with a slightly broader spread; it plateaus earlier but maintains good diversity.
Matrix	Moderate Exploitation	Performs reasonably but with a broader energy distribution; slower convergence than the best performers.
Multi Point	Moderate Exploration	Slightly more diverse offspring but slower in finding the global minimum; energy spread is larger than that of the top methods.
Three Parent	High Exploration (unstable)	The highest energy spread is poor convergence to low-energy structures due to excessive randomization.

Table 2: Summary of GA Crosssover Strategy Findings.

Chapter 5: Gaussian Process Acceleration

5.1 Integration with GA

A GP surrogate model was integrated into the GA workflow to enhance computational efficiency further and accelerate convergence. The revised procedure is illustrated in Figure 14.

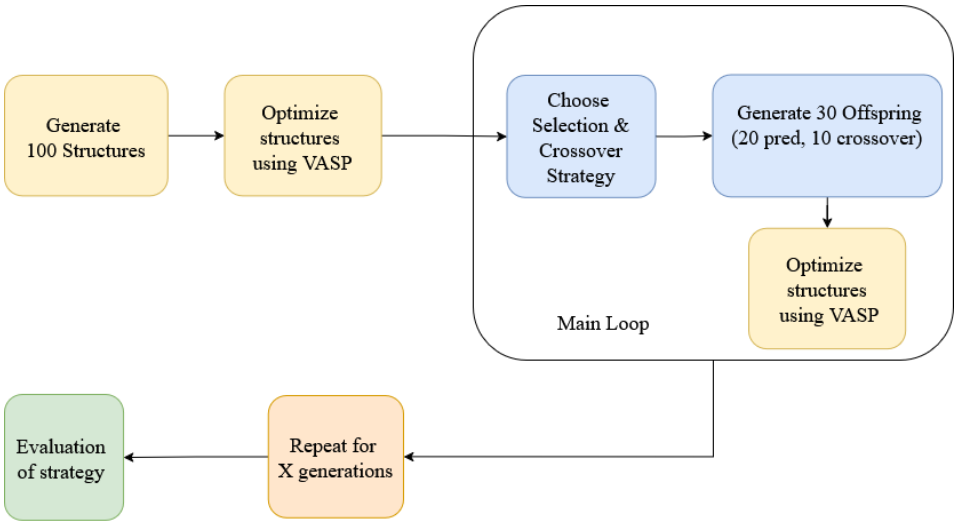


Figure 14: Workflow for Gaussian accelerated process.

The process begins with generating 100 initial structures, which are relaxed using VASP to obtain accurate DFT energy values. These relaxed structures form the initial training dataset for the GP model. Following this, the main loop of the genetic algorithm proceeds by choosing a selection and crossover strategy based on prior benchmarking results.The selected strategies are tournament selection and Half Uniform crossover. In each generation, 30 offspring structures are produced: 20 structures are predicted and selected directly using the GP model (by identifying candidates expected to have low energy). Compared to 10 structures are generated through traditional crossover operations applied to existing structures. This hybrid strategy combines the genetic algorithm's exploratory capabilities with the surrogate model's predictive efficiency. The newly created offspring, both GP-predicted and crossover-generated, are relaxed using VASP to validate their actual energies. The results from each generation are used to retrain and refine the GP model, progressively improving its predictive accuracy over time. This iterative cycle of prediction, variation, evaluation, and model retraining is repeated for up to 5 generations.

To assess the effectiveness of the Gaussian-accelerated approach, the results are compared directly against the GA-only approach, focusing on the lowest energy achieved. By comparing the minimum energy structures obtained by each method, the impact of GP acceleration on optimization success can be quantitatively evaluated. This hybrid strategy aims to significantly reduce computational cost while guiding the search toward global or near-global minima on the potential energy surface.

5.2 Results and Analysis

5.2.1 Feature Importance and Gaussian Process Hyperparameter Tuning for Slab Systems

Before integrating the Gaussian Process (GP) model into the genetic algorithm framework, an analysis of feature importance and hyperparameter sensitivity was performed to optimize the surrogate model's performance on the slab dataset. Figure 15 summarizes the results.

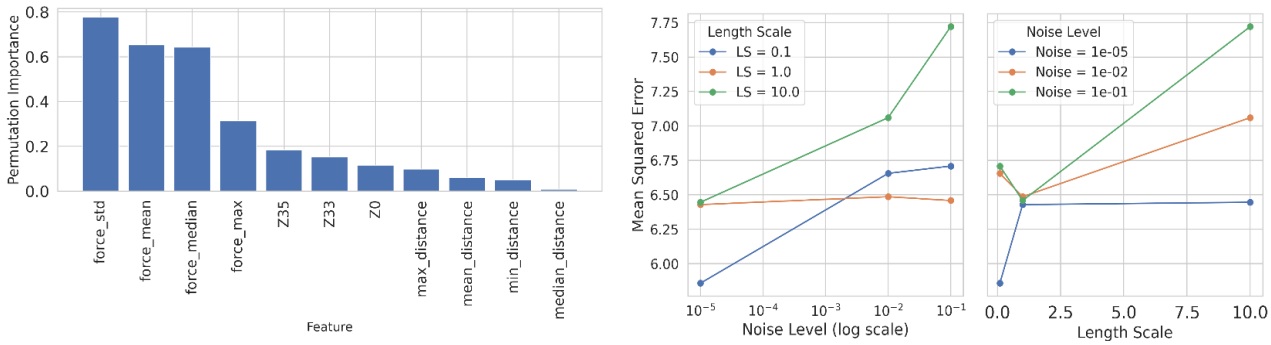


Figure 15:Feature importance and hyperparameter optimization summary. The left panel shows the relative importance of input features in predicting DFT energy, highlighting the most influential structural descriptors. The right panel presents the results of hyperparameter tuning, including optimal values that minimize model error and improve predictive performance.

The left panel shows the importance of the permutation-based feature for the GP model trained on slab structures. The most influential features were those derived from the atomic force information:

- force_ standard deviation, force_mean, and force_median emerged as the top three predictors, indicating that the statistical properties of atomic forces are highly correlated with total energy.
- The next most important features (Z23, Z33, and Z0) are the atomic numbers, highlighting the importance of chemical descriptors in capturing energy trends.
- Distance-based features such as max_distance, mean_distance, min_distance, and median distance contribute less strongly but still provide useful complementary information.

This shows that local force distributions play a critical role in accurately predicting the relaxed energies of slab structures.

The middle and right panels show the effect of length scale and noise level hyperparameters on the GP model's mean squared error (MSE).

- Length Scale (Middle Panel): Shorter length scales (e.g., LS=0.1) generally produced lower MSE, suggesting that energy depends sensitively on small variations in structural features. Models with longer length scales (LS=10.0) underperformed, likely due to over-smoothing.

- Noise Level (Right Panel):A lower noise level (1×10^{-5}) resulted in lower errors across different length scales. Higher noise (1×10^{-2}) resulted in higher MSE.

Overall, the best model performance was achieved using a small length scale (10) and a low noise level (1×10^{-5}) resulted in the lowest MSE of 0.448.

5.2.2 Feature Importance and Gaussian Process Hyperparameter Tuning for Cluster Systems

Similar to the slab system, the cluster dataset underwent feature importance analysis and Gaussian Process (GP) hyperparameter tuning to optimize surrogate modeling performance. Figure 16 summarizes these findings.

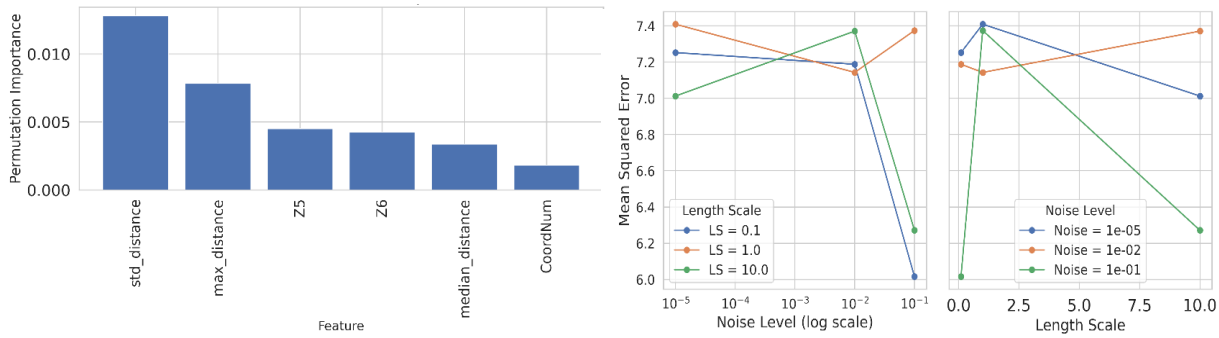


Figure 16:Feature importance and hyperparameter optimization summary. The left panel shows the relative importance of input features in predicting DFT energy, highlighting the most influential structural descriptors. The right panel presents the results of hyperparameter tuning, including optimal values that minimize model error and improve predictive performance.

The left panel shows the importance of the permutation-based feature for the cluster structures. Unlike slabs, the most critical features here are primarily distance-based descriptors:

- std_distance (standard deviation of pairwise atomic distances) emerges as the most important feature, indicating that variability in atomic spacing strongly correlates with cluster stability and total energy.
- max_distance also ranks highly, suggesting that the spatial extent of the cluster influences its energetic stability.

The middle and right panels present the sensitivity of the GP model's mean squared error (MSE) to changes in length scale and noise level.

- Length Scale (Middle Panel): A shorter length scale (LS=0.1) generally yielded the lowest MSE for clusters, suggesting that energy depends sharply on small changes in structural descriptors, similar to slabs. However, the performance gap across different length scales is somewhat narrower than that of slabs.

- Noise Level (Right Panel): A low noise level (1×10^{-5}) consistently resulted in better performance. Higher assumed noise levels degraded model accuracy, reinforcing that the training data for clusters is clean and benefits from a noise-free assumption.

Overall, the best model performance was achieved using a small length scale (10) and a low noise level (1×10^{-1}) resulted in the lowest MSE of 0165.

5.2.3 Final Structures: GA vs. Gaussian-Accelerated GA

Slab System (FCC(111))

The final structures and energy convergence behavior for the slab system under both traditional GA and GP-accelerated GA workflows are presented in Figure 17. The left structure corresponds to the best configuration obtained after five generations using a GA-only strategy with the Half Uniform crossover and tournament selction. The right structure shows the best candidate identified using the hybrid GA+GP approach.

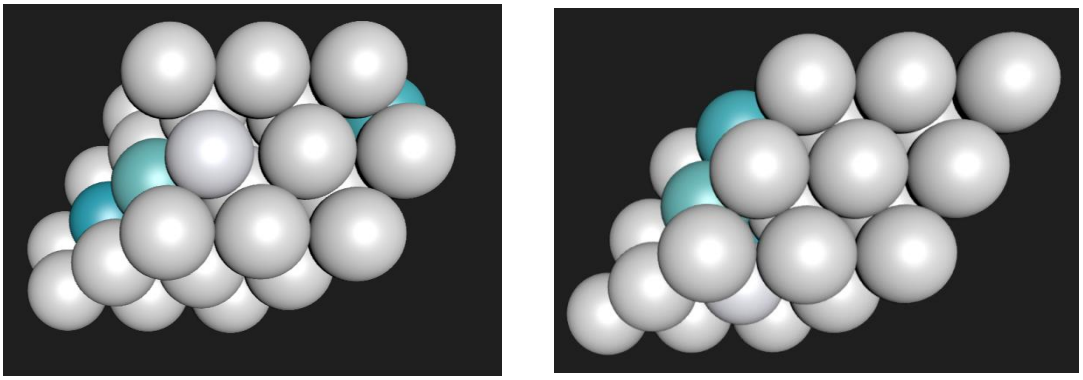


Figure 17: Comparison of final slab structures and energy convergence between GA-only and GA+Gaussian Process (GP)-accelerated optimization workflows.(a) The lowest-energy slab structure was identified using the GA-only approach after five generations, achieving a total energy of ***−93.530 eV****.(b) Best slab structure found by the GP-assisted GA in Generation 3, with an energy of* ***−93.447 eV****, demonstrating faster convergence*

In the GA-only workflow, the lowest energy achieved after five generations was −93.530 eV. By contrast, the GP-accelerated GA reached a near-optimal energy of −93.447 eV in just three generations, highlighting a significant improvement in convergence speed. While further generations in the GP-accelerated run resulted in slightly higher energies (e.g, −93.198 eV in Generations 4 and 5), this behavior reflects the model's exploratory nature and its tendency to sample uncertain or diverse regions of the search space.

Although the GP-accelerated method did not find a strictly lower minimum than the GA-only strategy, its ability to rapidly locate a highly competitive configuration illustrates the advantage of surrogate-assisted search. By focusing DFT evaluations on regions with high predicted utility, the GP model significantly reduces computational overhead while maintaining high-quality solutions.

This efficiency makes it particularly valuable for large-scale or time-constrained optimization of slab-based systems.

Cluster System (Ni_6Pd_4)

The final structures and energy convergence behavior for the cluster system under both traditional GA and GP-accelerated GA workflows are presented in Figure 18. The left structure corresponds to the best configuration obtained after three generations using a GA-only strategy with the Half Uniform crossover and tournament selection. The right structure shows the best candidate identified using the hybrid GA+GP approach found after one generation.

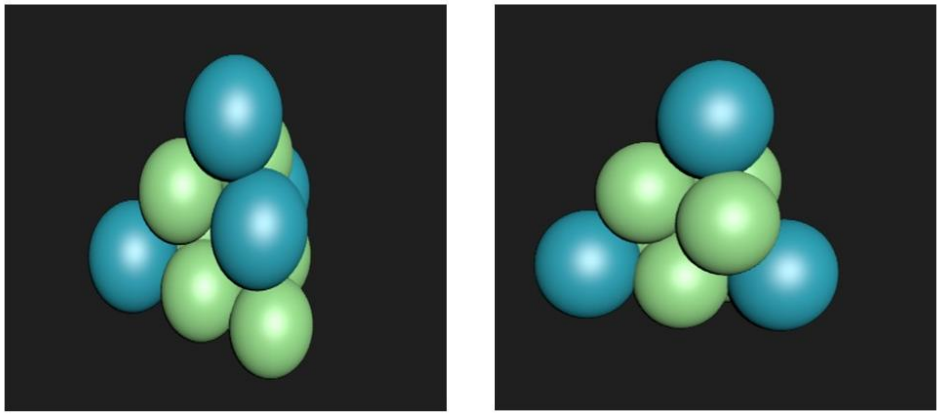


Figure 18: (a) Global optimum structure identified after Generation 1, with an energy of -35.051 eV.(b) Global optimum structure identified after Generation 3, with a slightly lower energy of -35.067 eV.Both structures exhibit the known global minimum octahedral geometry. However, in structure (b), the Pt atom appears slightly shifted to the left compared to structure (a), indicating a minor atomic rearrangement that results in improved stability.

Although both structures share the same overall geometry, a subtle difference is observed in (b): the Pt atom (depicted in blue) is slightly shifted to the left compared to its position in (a). This minor atomic rearrangement leads to a more stable configuration, demonstrating the genetic algorithm's ability to exploit small geometric variations for energy improvement.

This case highlights the importance of structural refinement across generations in cluster optimization. Unlike slabs, clusters possess a highly flexible geometry and a more complex potential energy surface, making such fine-scale optimizations critical. The observed improvement underscores the GA's effectiveness not just in exploring broad configurational spaces, but also in honing in on subtle but impactful structural changes that enhance overall stability.

Chapter 6: Conclusion & Future Work

6.1 Summary of findings

This study explored the optimization of atomic structures using GAs, focusing on selection strategies, crossover operators, and the integration of GP surrogate models to accelerate convergence. Key findings include:

- Ranking selection proved the most effective, consistently guiding the search toward lower-energy structures while maintaining population diversity. Tournament selection performed comparably but with slightly less exploration capability.
- Among crossover operators, Simple Cut Splice crossover and Half Uniform are the most effective, though they are statistically different, compared to Matrix, MultiPoint, and Three Parent crossovers.
- Integration of a Gaussian Process model significantly accelerated convergence, reaching near-optimal energies in fewer generations compared to the GA-only workflow and reducing computational costs.
- For slab systems, force-based descriptors were found to be the most important for energy prediction, while for cluster systems, distance-based features dominated.
- The hybrid GA-GP framework successfully identified global minimum configurations more efficiently, confirming the power of machine learning–accelerated optimization in atomistic simulations.

6.2 Limitations

While the study successfully demonstrated the potential of genetic algorithms and surrogate-assisted optimization for atomic structure discovery, several limitations were encountered that should be considered when interpreting the results:

- Queue Time OSCER: Access to computational resources at OSCER was subject to queue times, sometimes introducing significant delays between job submissions and completions. These delays limited the ability to conduct rapid iterative testing and forced prioritization of certain experiments over others. In particular, more extensive parameter searches such as varying crossover/mutation rates were constrained by resource availability.
- Limited Exploration of Selection Mechanisms: The study primarily compared traditional selection strategies such as tournament, ranking, elitism, and their hybrids. However, other sophisticated selection approaches were not tested such as Boltzmann selection, could offer a more adaptive exploration-exploitation balance.
- Simplification of Material Systems: The test cases, while representative, involved relatively small-scale slab and cluster models (e.g., 36 atoms for the slab system). Real-world materials often include hundreds or thousands of atoms, complex surfaces, and

heterogeneous compositions. The current framework may require further scalability and robustness enhancement to handle larger and chemically more complex systems efficiently.

- Gaussian Process Model Assumptions: The surrogate models assumed smooth energy landscapes and Gaussian noise properties. However, real material energy surfaces can exhibit sharp discontinuities that violate GP assumptions. Future work may require surrogate models capable of accurately capturing such complex energy behaviors.

6.3 Future work

Building upon the results and lessons from this study, several promising avenues for future research are proposed:

- Hybridization with Advanced Machine Learning Models:Future work could explore combining genetic algorithms not just with Gaussian Processes, but with neural networks, graph-based models, or reinforcement learning agents that actively learn optimal exploration-exploitation policies. These hybrid approaches could enhance both prediction fidelity and structural generation strategies.
- Expansion to Multi-Objective Optimization: Many real-world material design problems involve trade-offs between multiple properties, such as stability, catalytic activity, conductivity, or mechanical strength. Extending the GP accelerated framework to support multi-objective optimization would allow for the generation of Pareto-optimal fronts and enable the selection of candidates that balance multiple desirable traits rather than focusing solely on minimizing total energy.
- Application to Practical Materials Challenges: With further development, the methods presented here can be directly applied to urgent problems in catalyst discovery, battery material design, hydrogen storage, or nanostructured alloy optimization. Practical application will require adapting the framework to specific target properties and integrating domain-specific constraints (e.g. stability under operating conditions).
- Automate the entire workflow from structure generation and relaxation to energy extraction and dataset preparation, and integrate a database system (e.g., MongoDB or SQLite) to store metadata and simulation outputs, enabling streamlined experimentation, reduced human error, and efficient result retrieval and comparison.

Works Cited

[1] A. Gad, “Adaptive Mutation in Genetic Algorithm With Python Examples,” neptune.ai. Accessed: Jan. 28, 2025. [Online]. Available: https://neptune.ai/blog/adaptive-mutation-in-genetic-algorithm-with-python-examples

[2] C. J. Pickard and R. J. Needs, “Ab initio Random Structure Searching,” *J. Phys. Condens. Matter*, vol. 23, no. 5, p. 053201, Feb. 2011, doi: 10.1088/0953-8984/23/5/053201.

[3] “Stochastic Algorithm - an overview | ScienceDirect Topics.” Accessed: Apr. 29, 2025. [Online]. Available: https://www.sciencedirect.com/topics/computer-science/stochastic-algorithm

[4] J. P. K. Doye, M. A. Miller, and D. J. Wales, “Evolution of the potential energy surface with size for Lennard-Jones clusters,” *J. Chem. Phys.*, vol. 111, no. 18, pp. 8417–8428, Nov. 1999, doi: 10.1063/1.480217.

[5] C. Barrón, S. Gómez, D. Romero, and A. Saavedra, “A genetic algorithm for Lennard-Jones atomic clusters,” *Appl. Math. Lett.*, vol. 12, no. 7, pp. 85–90, Oct. 1999, doi: 10.1016/S0893-9659(99)00106-8.

[6] B. Lei *et al.*, “Bayesian optimization with adaptive surrogate models for automated experimental design,” *Npj Comput. Mater.*, vol. 7, no. 1, pp. 1–12, Dec. 2021, doi: 10.1038/s41524-021-00662-x.

[7] V. L. Deringer, A. P. Bartók, N. Bernstein, D. M. Wilkins, M. Ceriotti, and G. Csányi, “Gaussian Process Regression for Materials and Molecules,” *Chem. Rev.*, vol. 121, no. 16, pp. 10073–10141, Aug. 2021, doi: 10.1021/acs.chemrev.1c00022.

[8] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2005. doi: 10.7551/mitpress/3206.001.0001.

[9] E. Aarts, J. Korst, and W. Michiels, “Simulated Annealing,” in *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*, E. K. Burke and G. Kendall, Eds., Boston, MA: Springer US, 2005, pp. 187–210. doi: 10.1007/0-387-28356-0_7.

[10] “Kohn-Sham Theory - an overview | ScienceDirect Topics.” Accessed: Apr. 29, 2025. [Online]. Available: https://www.sciencedirect.com/topics/chemistry/kohn-sham-theory

[11] D. Waroquiers *et al.*, “*ChemEnv* : a fast and robust coordination environment identification tool,” *Acta Crystallogr. Sect. B Struct. Sci. Cryst. Eng. Mater.*, vol. 76, no. 4, pp. 683–695, Aug. 2020, doi: 10.1107/S2052520620007994.

[12] “GaussianProcessRegressor,” scikit-learn. Accessed: Mar. 03, 2025. [Online]. Available: https://scikit-learn/stable/modules/generated/sklearn.gaussian_process.GaussianProcessRegressor.html

[13] “WhiteKernel,” scikit-learn. Accessed: Apr. 04, 2025. [Online]. Available: https://scikit-learn/stable/modules/generated/sklearn.gaussian_process.kernels.WhiteKernel.html

[14] R. K. Raju, S. Sivakumar, X. Wang, and Z. W. Ulissi, “Cluster-MLP: An Active Learning Genetic Algorithm Framework for Accelerated Discovery of Global Minimum Configurations of Pure and Alloyed Nanoclusters,” *J. Chem. Inf. Model.*, vol. 63, no. 20, pp. 6192–6197, Oct. 2023, doi: 10.1021/acs.jcim.3c01431.

[15] B. L. Miller, “Genetic Algorithms, Tournament Selection, and the Effects of Noise”.

[16] “Tournament Selection (GA),” GeeksforGeeks. Accessed: Apr. 29, 2025. [Online]. Available: https://www.geeksforgeeks.org/tournament-selection-ga/

[17] A. Garg, “Crossover Operators in Genetic Algorithm,” Geek Culture. Accessed: Jan. 28, 2025. [Online]. Available: https://medium.com/geekculture/crossover-operators-in-ga-cffa77cdd0c8
[18] sonia rani, “I am a beginner in DFT. What are some resources that could help me to learn the basics?,” Matter Modeling Stack Exchange. Accessed: Mar. 03, 2025. [Online]. Available: https://mattermodeling.stackexchange.com/q/1046
[19] L. B. Vilhelmsen and B. Hammer, “Systematic Study of Au 6 to Au 12 Gold Clusters on MgO(100) F Centers Using Density-Functional Theory,” *Phys. Rev. Lett.*, vol. 108, no. 12, p. 126101, Mar. 2012, doi: 10.1103/PhysRevLett.108.126101.
[20] L. B. Vilhelmsen and B. Hammer, “A genetic algorithm for first principles global structure optimization of supported nano structures,” *J. Chem. Phys.*, vol. 141, no. 4, p. 044711, Jul. 2014, doi: 10.1063/1.4886337.
[21] G. Rossi and R. Ferrando, “Searching for low-energy structures of nanoparticles: a comparison of different methods and algorithms,” *J. Phys. Condens. Matter*, vol. 21, no. 8, p. 084208, Jan. 2009, doi: 10.1088/0953-8984/21/8/084208.
[22] D. Waroquiers *et al.*, “Statistical Analysis of Coordination Environments in Oxides,” *Chem. Mater.*, vol. 29, no. 19, pp. 8346–8360, Oct. 2017, doi: 10.1021/acs.chemmater.7b02766.
[23] S. Han, G. Barcaro, A. Fortunelli, S. Lysgaard, T. Vegge, and H. A. Hansen, “Unfolding the structural stability of nanoalloys via symmetry-constrained genetic algorithm and neural network potential,” *Npj Comput. Mater.*, vol. 8, no. 1, p. 121, Jun. 2022, doi: 10.1038/s41524-022-00807-6.
[24] A. Granja-DelRío, H. A. Abdulhussein, and R. L. Johnston, “DFT-Based Global Optimization of Sub-nanometer Ni–Pd Clusters,” *J. Phys. Chem. C*, vol. 123, no. 43, pp. 26583–26596, Oct. 2019, doi: 10.1021/acs.jpcc.9b05970.
[25] J. P. Perdew, K. Burke, and M. Ernzerhof, “Generalized Gradient Approximation Made Simple,” *Phys. Rev. Lett.*, vol. 77, no. 18, pp. 3865–3868, Oct. 1996, doi: 10.1103/PhysRevLett.77.3865.
[26] “Global minimum structure searches via particle swarm optimization - Call - 2007 - Journal of Computational Chemistry - Wiley Online Library.” Accessed: Apr. 28, 2025. [Online]. Available: https://onlinelibrary.wiley.com/doi/full/10.1002/jcc.20621

Appendix

Appendix A: Sample CSV Data from Different Strategies

Strategy	Generation	Structure_ID	Energy	Parent
ranking	GA_1	image_1	-93.345	Self
tournament	GA_1	image_1	-88.567	Parent 1
elitism	GA_1	image_10	-91.234	Parent 2

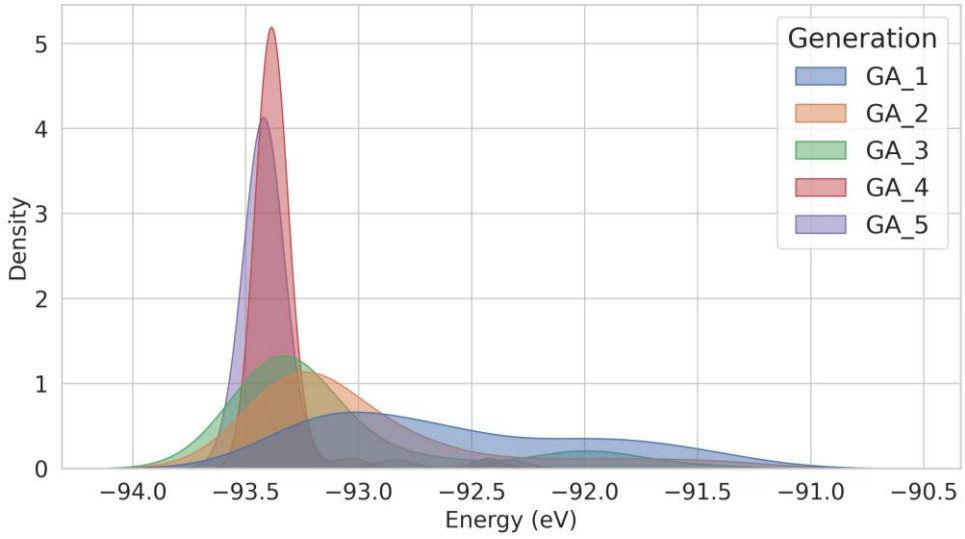
Appendix B: Sample CSV Data of Parent Structures

Parent	Energy	Traj location
Parent 1	-93.123	/ourdisk/hpc/gunasoorya/2
Parent 2	-92.955	/ourdisk/hpc/gunasoorya/1

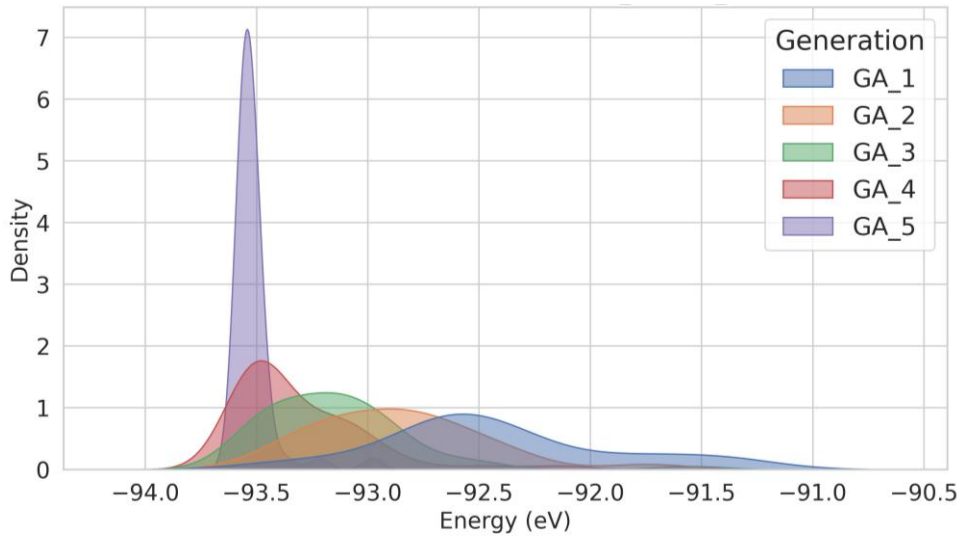
Appendix C: Sample Feature Dataset for Machine Learning Model

Structure_ID	Energy	Mean_dist	Min_dist	Max_dist	Z0	Z9	Force_mean	Force_min	Force_max	Co-number
image_1	-33.345	4.437	2.243	6.945	28	46	0.017	0.001	0.034	16
image_10	-34.567	4.051	1.967	7.678	46	28	0.031	0.002	0.059	15.6
image_18	-34.234	4.345	2.56	7.012	46	28	0.103	0.005	0.067	14.2

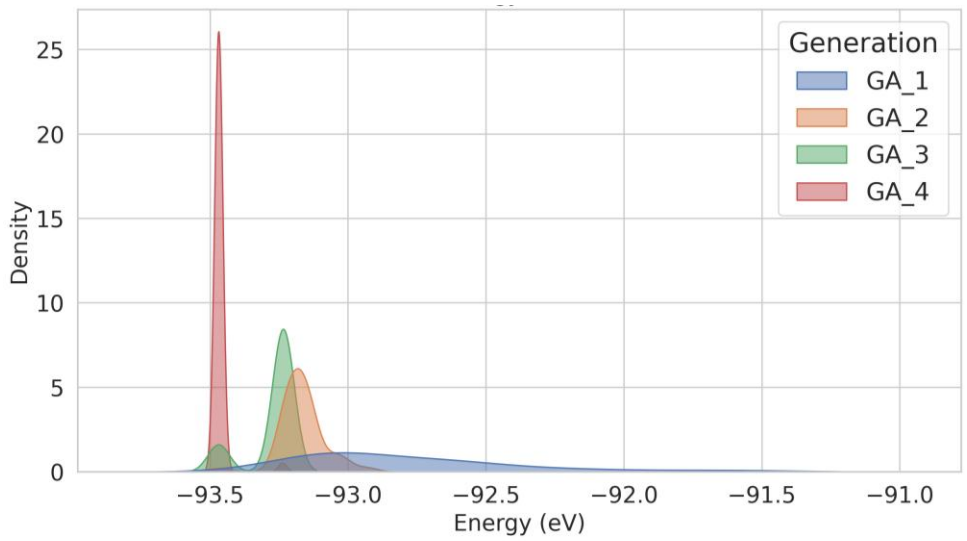
Appendix D: Energy Distribution Across Generations for Tournament Selection



Appendix E: Energy Distribution Across Generations for Ranking Selection



Appendix F: Energy Distribution Across Generations for Half Uniform Crossover (GA1 – GA4)



Appendix G: Energy Distribution Across Generations for Half Uniform Crossover (GA1 – GA5)