Diamond Energy – a systematic conformation searching method

Mengman Wei

Supervisor: Prof. J. M. Goodman

Department of Chemistry
University of Cambridge

This dissertation is submitted for the degree of
Doctor of Philosophy

Newnham College January 2024

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Mengman Wei
January 2024

Abstract

Fully understanding the properties of molecules requires a good knowledge of all low-energy conformations. A python program, *Diamond Energy*, was developed based on an assumption that the structures of local minima are similar to a diamond lattice. These conformers could be found by calculations with integer arithmetic and evaluated with simple energy equations. Both of these should speed up systematic searching. Test results on alkane showed that *Diamond Energy* was able to do systematic conformational searching for acyclic alkanes exhaustively, accurately and very fast.

The scope of the program's application was extended beyond acyclic alkanes to include six-membered rings, and then further extended to include saccharides. This extension was facilitated by the development of an automatic pipeline for searching suitable energy evaluation parameters when introducing new atom types.

Finally, using the large quantity of data generated by the *Diamond Energy* test process, a machine learning pipeline was applied and fed with *Diamond Energy* data. This made it possible to explore the geometry distribution of diamond lattice framework and help to improve the conformational searching process.

Acknowledgements

I am deeply grateful to Professor Jonathan for his unwavering guidance, patience, and support throughout my PhD journey. The foundation of this project rests on his groundbreaking work, the *Diamond Energy* algorithm. His vast knowledge and sophistication have been pivotal, and every discussion with him has been a learning experience. I am honored to have studied under such a revered scholar. He generously allowed me into his group and provided consistent guidance. Although I began with a limited background in computational chemistry and produced work that wasn't up to standard initially, his patience never waned. He consistently directed my efforts toward meaningful research. Admittedly, I sometimes came across as abrupt and impolite; for that, I sincerely apologize. Nevertheless, Professor Jonathan persevered with me, exemplifying what it means to be an advisor in academic research. He does not just hand out answers; instead, he encourages independent discovery. This approach fosters genuine understanding and nurtures research skills. My time under his mentorship has been transformative.

I would also like to extend my gratitude to the members of the Goodman Group for their enlightening discussions, suggestions, and encouragement. My appreciation goes to my first-year viva examiners, Professor David Spring and Professor Nick Bampos, for their invaluable feedback. Professor Spring, as my graduate advisor, has been supportive, especially during the challenging COVID period. I'm grateful to the technical and library staff, as well as the entire department and college teams, for their invaluable assistance.

Cambridge is one of the best places in the world to study and do research. I sincerely appreciate that I could have a chance to study here. The university is a hub of brilliance, and the individuals I've encountered here have greatly influenced my outlook. Inspired by them, I'm driven to better myself, to embody politeness, dignity, and integrity. The wealth of academic resources at Cambridge is unparalleled, and I regret not utilizing them more extensively earlier on. This experience will remain etched in my memory, reminding me that learning and research are lifelong endeavors.

I am sincerely appreciative for the opportunity to study in the UK, a nation renowned for its contributions to science and democracy—a beacon of human civilization. This experience has significantly broadened my horizons, allowing me to understand the essence of a healthy and civil society. I deeply appreciate the wonderful individuals I've encountered here and am continuously inspired to become a better person.

I must also express my heartfelt gratitude to my family, particularly my mother. Her unwavering belief in me and her endless love and encouragement have been my anchors.

Time has flown by so quickly, and as my Cambridge chapter concludes, I'm acutely aware of the many things left unexplored. I depart with profound appreciation, carrying with me the determination to continue my journey of growth and discovery.

Table of contents

List of figures **xiii**

List of tables **xxi**

Nomenclature **xxix**

1 Introduction **1**

1.1 Understanding the Crucial Role of Conformation Searching in Molecular Modeling 2

1.2 Conformation Searching Algorithms Survey 3

1.2.1 Global Search for Molecular Structures Generation 4

1.2.1.1 Build-Up Approaches 4

1.2.1.2 Systematic Methods 5

1.2.1.3 Random Transformations 10

1.2.1.4 Genetic Algorithm (GA) 12

1.2.1.5 Molecular Dynamics (MD) 13

1.2.1.6 Distance Geometry (DG) 17

1.2.1.7 Rule-Based and Knowledge-Based Strategies 18

1.2.1.8 Emerging Machine Learning (ML) Methods 19

1.2.2 Local Minimization for Structures Fine Tuning 21

1.2.2.1 Empirical Force field – Molecular Mechanics (MM) . . . 21

1.2.2.2 Quantum Mechanics (QM) 22

1.2.2.3 Solvent Models 22

1.2.2.4 Energy Minimized Algorithms 23

1.2.3 The Evaluation of a Conformational Searching 24

1.3 Conclusions 25

2 *Diamond Energy* Method **27**

2.1 *Diamond Energy* Introduction 27

2.2 Computational Details 32
2.3 Results and Discussion 40
2.3.1 Evaluation Dataset for the *Diamond Energy* Algorithm 40
2.3.2 Comprehensive Analysis and Demonstration of *Diamond Energy*'s Operational Mechanics 40
2.3.3 Assessing *Diamond Energy*'s Overall Speed and Capability for Geometry Creation 63
2.3.4 Evaluating *Diamond Energy*'s Conformer Search Capabilities Against MCMM, SPMC, LMCS, and MCLMCS under MM3/OPLS4 65
2.3.4.1 An Overall Comparative Analysis of *Diamond Energy*'s Accuracy 67
2.3.4.2 Completeness in Conformer Sampling by *Diamond Energy* 76
2.3.4.3 Computational Efficiency: Assessing the Running Speed of *Diamond Energy* 85
2.3.5 Case Study: Utilizing *Diamond Energy* to Interpret Experimental Data 89
2.4 Conclusions 93

3 Further Development of *Diamond Energy* 95
3.1 Incorporation of Cyclohexyl Groups into the Model 98
3.2 Incorporating Oxygen Atoms into the Model 103
3.2.0.1 Establishing Approximate Ranges for New Parameters . . 104
3.2.0.2 Employing Bayesian Optimization to Determine Potential Parameter Combinations Within the Defined Range . . . 109
3.3 *Diamond Energy* Further Development - Results and Conclusions 119
3.3.1 A Detailed Analysis and Illustration of *Diamond Energy* II Results . 120
3.3.2 A Comparative Analysis the Accuracy of *Diamond Energy* II 123
3.3.2.1 Identifying the Lowest Energy in the Test Set Using *Diamond Energy* II 123
3.3.2.2 Determining Conformers Within 3 to 5 kcal/mol of the Minimum Energy Using *Diamond Energy* II in the Test Set 132
Conclusions 134
3.3.2.3 A Comparative Analysis the Computational Efficiency of *Diamond Energy* II 136
Conclusions 150
3.4 Conclusions 151

4 *Diamond Energy* Extension Work – Diamond AI 153
4.1 Leveraging *Diamond Energy* Data for the Accelerated Identification of Low-Energy Conformers Through Machine Learning 154
4.1.1 Background 154
4.1.2 Results and Discussion 161
4.1.2.1 Experimental Workflow 161
4.1.2.2 Dataset 161
4.1.2.3 Results and Discussion 166
Assessing the Performance of Conformer Generation Models: CGCF, GeoMol, and ConfVAE in Learning Diamond Lattice Low Energy Conformer Distributions 166
Evaluating Model Performance: The Impact of Training Data on ML-Based Models and Their Subsequent Comparison with the *Diamond Energy* Algorithm in Real-World Conformational Search Tasks 168
Variations in Training Data (Diamond, ISO17, QM9, DRUGs) and Their Impact on the Output of CGCF model 168
Variations in Training Data (Diamond, QM9, Drugs) and Their Impact on the Output of GeoMol Model . 178
Diamond Training Data on ConfVAE Model VS *Diamond Energy* Algorithm 184
GraphDG Results 186
Conclusions 186
Exploring the Potential for Accelerated Conformer Minimization: Machine Learning Models vs. PRDG . . . 187
Conclusions 194
4.2 Conclusions 194

5 Conclusions 197

References 199

Appendix A 209

List of figures

1.1 The sequence of transformations for butane. 18

2.1 In this diamond lattice-based model, low-energy conformers can be rapidly pinpointed (Taking pentane as an example). 28
2.2 Example of a pentane searching tree in *Diamond Energy*. 30
2.3 Comprehensive Representation of Non-Bonded Interactions within Diamond Lattice:This figure provides an in-depth visual of the various non-bonded interactions found in the diamond lattice's conformers. Each interaction type, from 'Overlap' to 'Axeq (axial and equatorial positions)', is categorized based on non-bonded atomic distances, and their respective energy contributions are highlighted. The terminal atoms are denoted as **a** and **b**, and the distances of their non-bonded interactions are distinctly marked in green. This illustration serves as a detailed guide, emphasizing the roles of these interactions in dictating the strain energy of molecular structures within the lattice. 32
2.4 Flowchart of *Diamond Energy*. 33
2.5 In the depicted lattice, each atom has the potential to form up to four covalent bonds. These bonds can be oriented either towards or away from a designated reference point, labeled as center **c**. 35
2.6 Matrix of eight direction vectors for each step between atoms, reflecting the potential orientations of covalent bonds in the diamond lattice. 35
2.7 A look-up table for fast calculating the true direction of the current working atom. 36
2.8 Illustrates the process employed in the *Diamond Energy* algorithm to generate and calculate the geometry of new conformers. This is done by utilizing a lookup table and a matrix of eight directional vectors. 36
2.9 Molecules used for testing the performance of *Diamond Energy*. 41
2.10 Molecule **c4** example. 42

2.11 Process of Constructing Initial Structures for the **c4** Molecule: First Three Atoms Form a Fixed Propane Coordinate(a), with the Remaining Atom Extending Along the Propane Chain on a Diamond Lattice(b). The Coordinates of the Atoms Are Listed in the Brackets Beside Them. 42
2.12 Conformer Generation in the *Diamond Energy* Algorithm for Molecule **c4**: Rotatable Bond in Red, Rotation Direction in Green, and Atom Affected by the Twist Highlighted in Blue. The Coordinates of the Atoms Are Listed in the Brackets Beside Them. 43
2.13 Evaluation of Conformer Energies from the *Diamond Energy* Algorithm for Molecule **c4**. (Rotatable bond: Red; Rotation direction: Green; Atoms affected by twist: Highlighted in blue; Non-bond interactions: Indicated by arrows with atom pairs, distances, and interaction types labeled in red; Atom coordinates: Provided in brackets next to them). 44
2.14 Three Low Energy Conformers in SDF Format Obtained Directly from the *Diamond Energy* Algorithm for Molecule **c4**. 44
2.15 Three Low Energy Conformers Obtained from the *Diamond Energy* Algorithm for Molecule **c4** After Adding Hydrogen by RDKit in SDF Format. . 46
2.16 The analysis revealed three conformers of **c4** as identified by *Diamond Energy*, which were then compared to those found by MCLMCS. On the left, we have configurations from *Diamond Energy*, and on the right, the corresponding configurations from MCLMCS. The key difference between the *Diamond Energy* and MCLMCS results, particularly for the anti configuration of **c4** , lies in the orientation of the hydrogen atoms attached to the terminal carbons. The calculation were done in OPLS4 force field. 47
2.17 Molecule **c5** example. 49
2.18 Sequence of Steps for *Diamond Energy* to Construct the **c5** Structure within a Diamond Lattice, Following InChI Connectivity Information. InChI Atoms involved in each step are highlighted in red. 50
2.19 Molecule **c5n** example. 50
2.20 Illustrating the method employed by *Diamond Energy* for conducting a conformer search on the **c5n** molecule (pentane) by utilizing a tree search algorithm. 51
2.21 Illustrating the method employed by *Diamond Energy* for conducting a conformer search on the **c5n** molecule (pentane) by utilizing a tree search algorithm. 51

2.22 Illustrating the method employed by *Diamond Energy* for conducting a conformer search on the **c5n** molecule (pentane) by utilizing a tree search algorithm. 52
2.23 Molecule **c8a** and **c8s** examples. 55
2.24 This figure illustrates the molecules **c8a** and **c8s**, which are identical except for two opposite stereocenters. The program's default configuration initially assigns the same structure to both molecules. This leads to an accurate structure for **c8s**, but an error in the stereocenter for **c8a**. Consequently, **c8s** moves forward to the conformer search phase, as shown in previous examples, whereas **c8a** requires adjustments to correct the stereocenter discrepancy. . 56
2.25 Total number of structures generated and total number of low energy conformers output by *Diamond Energy* algorithm. 64
2.26 Total running time and average running time by *Diamond Energy* algorithm. 65
2.27 Comparative Workflow of *Diamond Energy* Algorithm and Four Leading Conformational Search Algorithms with Energy Minimization and Frequency Analysis. 66
2.28 This figure shows a side-by-side comparison of the lowest energy conformers identified by various computational approaches using the MM3 force field. (The data here suggest that, for most of the test scenarios utilizing the MM3 force field, there is a consensus among the different methods regarding the identification of the conformer with the lowest energy. However, in certain cases, specifically with molecules **c5**, **c8**, **c12p**, **c12p0**, **c12p1**, and **c26**, *Diamond Energy* reported slightly higher energy values for the lowest-energy conformers. Further investigation into the molecular structures revealed that *Diamond Energy* does, in fact, recognize the same lowest-energy carbon chains as identified by the comparative methods. A detailed explanation is provided in the following paragraph to offer additional insight into this observation. 67
2.29 RMSD Values Comparing the Lowest-Energy Conformer from *Diamond Energy* and Other Methods in MM3 Force Field. 69
2.30 Comparative Analysis of Lowest Energy Conformers Using Distinct Hydrogen Treatment Techniques of *Diamond Energy* Prior to Optimization with MacroModel. 70
2.31 This figure shows a side-by-side comparison of the lowest energy conformers identified by various computational approaches using the MM3 force field. . 70
2.32 The lowest energy levels box plot compare across different methods (MM3). 71

2.33 Comparison of the Minimum Energy Values Obtained Through Different Computational Methods Using the OPLS4 Force Field. 75
2.34 The lowest energy levels box plot compare across different methods (OPLS4). 76
2.35 The total execution time required to generate all molecular structures for a test molecule of *Diamond Energy* and other methods. 86
2.36 The average time needed to produce a single structure of *Diamond Energy* and other methods. 87
2.37 Total Execution Time for Conformer Search Comparing *Diamond Energy* and Other Methods Using the MM3 Force Field. 87
2.38 Average Execution Time for Conformer Search Using *Diamond Energy* and Other Methods with the MM3 Force Field. 88
2.39 Total Execution Time for Conformer Search Comparing *Diamond Energy* and Other Methods Using the OPLS4 Force Field. 88
2.40 Average Execution Time for Conformer Search Using *Diamond Energy* and Other Methods using the OPLS4 Force Field. 89
2.41 Comparative analysis of molecules investigated in Aggarwal et al.'s study (top) and molecules subjected to *Diamond Energy* evaluation (bottom). . . . 90
2.42 Key Findings from the Study by Aggarwal et al [1]. 91
2.43 Right helix: Left helix ratio in three *Diamond Energy* conformer generation results. 92
2.44 Total helix ratio in three *Diamond Energy* conformer generation results. . . 93

3.1 Extending *Diamond Energy* to Complex Saccharide Systems: Structural Affinities with Diamond Lattices. 96
3.2 *Diamond Energy*: Upgrades and New Features in Version II. 97
3.3 An example of how *Diamond Energy* II builds the structure of methylcyclopentane. 99
3.4 Illustration of *Diamond Energy* II Variables and Their Adaptations to Generate New Conformers. 101
3.5 Illustration of *Diamond Energy* II Checking and Correcting Stereocenters. . 102
3.6 *Diamond Energy* II searching tree example. 103
3.7 Assumed Non-Bonded Interactions for Newly Introduced Oxygen Atoms. . 104
3.8 Molecules Selected for Testing Approximate Parameter Ranges. 105
3.9 Identification of a Different Interaction Motif in C-O Example. 106
3.10 Test Molecule Set of O-O Interaction Parameters. 108
3.11 SMILES Representations of Molecules Used to Test the Broad Range of New Energy Terms Introduced with Alcohol Oxygen. 109

3.12 Flowchart Depicting the Process of Parameter Set Optimization Based on Bayesian Optimization. 111
3.13 Motif Frequency Distribution Analysis of the Test Set for Ether Oxygen Related Parameters. 113
3.14 Hyperparameters Verifying Dataset Similarity Analysis for Ether Oxygen Related Parameters. 113
3.15 A diagram showing the relationship between Diamond Unique (purple), Maestro Unique (orange), and the Baseline. The evaluation score used in this study will utilize the sum of Diamond Unique and Maestro Unique as the best approximation for calculation. 115
3.16 Molecules used for testing the performance of *Diamond Energy* II Further development. 120
3.17 Molecules used for testing the performance of *Diamond Energy* II Further development. 121
3.18 Comparative Workflow of *Diamond Energy* Version II and Four Leading Conformational Search Algorithms with Energy Minimization and Frequency Analysis. 122
3.19 Molecule **e4** example. 122
3.20 Molecule **e4** example - *Diamond Energy* II read the information of its input process. 122
3.21 Molecule **e4** example - *Diamond Energy* II flip the ring feature. 123
3.22 A boxplot illustrating the distribution of lowest energy values for all molecules in the test set, across the six pipelines, using the MM3 force field. 126
3.23 A scatter plot that showcases the minimum energy levels for each molecule in the test set, as obtained by all six pipelines using the MM3 force field. . . 127
3.24 A boxplot illustrating the distribution of lowest energy values for all molecules in the test set, across the six pipelines, using the OPLS4 force field. 128
3.25 Displays a scatter plot that showcases the minimum energy levels for each molecule in the test set, as obtained by all six pipelines using the OPLS4 force field. 130
3.26 The total time cost to generate conformers of various methods: *Diamond Energy* II, SPMC, MCMM, LMCS, MCLMCS. 142
3.27 The average time cost to generate one conformer of various methods: *Diamond Energy* II, SPMC, MCMM, LMCS, MCLMCS. 143

3.28 The total running time cost to do a complete conformer search for all the test molecule by various methods: *Diamond Energy* II two pipelines, SPMC, MCMM, LMCS, MCLMCS, using MM3. 143
3.29 Average Running Time Cost of Each Method to Generate Conformers for All Test Cases Using MM3 Calculations. 148
3.30 Total Running Time Cost (Logarithmic Scale) of Each Method to Generate Conformers for All Test Cases Using OPLS4 Calculations. 148
3.31 Average Running Time Cost (Seconds) of Each Method to Generate Conformers for All Test Cases Using OPLS4 Calculations. 149

4.1 Workflow chart illustrating the use of *Diamond Energy* output data to train the current conformer search machine learning model. 162
4.2 *Diamond Energy* dataset used in this study. 163
4.3 *Diamond Energy* dataset used in this study. 164
4.4 *Diamond Energy* dataset used in this study. 165
4.5 *Diamond Energy* dataset used in this study. 165
4.6 Flow Chart Comparing the Effects of Different Datasets on Conformer Search Generation Models and Subsequent Comparison with *Diamond Energy*. . . 169
4.7 The results from Table 4.6, showing the count of conformers within 3 kcal/mol of the lowest energy obtained using various methods with MM3 calculations. 173
4.8 Visualization of the results from Table 4.7, showing the count of conformers within 5 kcal/mol of the lowest energy obtained using various methods with MM3 calculations. 174
4.9 The results from Table 4.8, showing the count of conformers within 3 kcal/mol of the lowest energy obtained using various methods with OPLS4 calculations. 175
4.10 Visualization of the results from Table 4.9, showing the count of conformers within 5 kcal/mol of the lowest energy obtained using various methods with OPLS4 calculations. 176
4.11 The results from Table 4.13, showing the count of conformers within 3 kcal/mol of the lowest energy obtained using various methods with MM3 calculations. 180
4.12 The results from Table 4.14, showing the count of conformers within 5 kcal/mol of the lowest energy obtained using various methods with MM3 calculations. 181

4.13 The results of the count of conformers within 3 kcal/mol of the lowest energy obtained using various methods with OPLS4 calculations. 182
4.14 The results of the count of conformers within 5 kcal/mol of the lowest energy obtained using various methods with OPLS4 calculations. 183
4.15 The results of the final minimization energy value of GeoDiff, ConfGF, and Diamond-PRCG using MM3. 193
4.16 The results of the final minimization energy value of GeoDiff, ConfGF, and Diamond-PRCG using OPLS4. 193

List of tables

2.1 Comparison of the energy values (kcal/mol) for the three low-energy conformers of **c4** identified using the *Diamond Energy* and MCLMCS algorithms with the OPLS4 force field. The table is organized as follows: the first column lists the conformer; the second and third columns present the energy values obtained by the respective methods; and the fourth and fifth columns show their RMSD values, with and without hydrogen comparison, respectively. 46

2.2 Comparison of the energy values (kcal/mol) for the three low-energy conformers of **c4** identified using the *Diamond Energy* and MCLMCS algorithms with the MM3 force field. The table is organized as follows: the first column lists the conformer; the second and third columns present the energy values obtained by the respective methods; and the fourth and fifth columns show their RMSD values, with and without hydrogen comparison, respectively. . 48

2.3 Comparison of **c4** MM3 energy of the conformers obtained by *Diamond Energy* using ETKDG optimizing the final structures and MCLMCS Algorithm. 49

2.4 Comparison of **c4** OPLS4 energy of the conformers obtained by *Diamond Energy* using ETKDG optimizing the final structures and MCLMCS Algorithm. 49

2.5 Comparison of the energy values (kcal/mol) for the three low-energy conformers of **c5** identified using the *Diamond Energy* and MCLMCS algorithms with the MM3 force field. The table is organized as follows: the first column lists the conformer; the second and third columns present the energy values obtained by the respective methods; and the fourth and fifth columns show their RMSD values, with and without hydrogen comparison, respectively. . 49

2.6 Results of **c5n** found by *Diamond Energy* and MCLMCS in MM3 (kcal/mol). 54

2.7 Results of **c5n** found by *Diamond Energy* and MCLMCS in OPLS4 (kcal/mol). 54

2.8 The MM3 results of **c8a** by *Diamond Energy* and MCLMCS (kcal/mol). . . 57

2.9 The OPLS4 results of **c8a** by *Diamond Energy* and MCLMCS (kcal/mol). . 57

2.10 The MM3 results of **c8s** by *Diamond Energy* and MCLMCS (kcal/mol). . . 58

2.11 The OPLS4 results of **c8s** by *Diamond Energy* and MCLMCS (kcal/mol). . 58
2.12 The MM3 energy values in kcal/mol for each conformation of **c4** found by *Diamond Energy* using RDKit hydrogen postprocessing pipeline, *Diamond Energy* using MacroModel hydrogen postprocessing pipeline and MCLMCS (kcal/mol). 60
2.13 RMSD Comparison for Each Conformation of **c4** Identified by *Diamond Energy* Using Different Hydrogen Addition Strategies and Compared with MCLMCS in MM3 calculations. This table compares the three low-energy conformers of **c4** as identified by *Diamond Energy* through two pipelines, alongside their RMSD comparisons with MCLMCS results. The table is organized as follows: the first column lists the conformer; the second and third columns display the RMSD values of the RDKit post-processing results in comparison with MCLMCS results, with and without the addition of hydrogen, respectively; the fourth and fifth columns present the RMSD values of the MacroModel post-processing results, with and without hydrogen comparison with MCLMCS, respectively. 60
2.14 The MM3 energy values in kcal/mol for each conformation of **c5** found by *Diamond Energy* RDKit post-processing pipeline, *Diamond Energy* MacroModel post-processing pipeline and MCLMCS (kcal/mol). 61
2.15 RMSD Comparison for Each Conformation of **c5** Identified by *Diamond Energy* Using Different Hydrogen Addition Strategies and Compared with MCLMCS in MM3 calculations. This table compares the three low-energy conformers of **c5** as identified by *Diamond Energy* through two pipelines, alongside their RMSD comparisons with MCLMCS results. The table is organized as follows: the first column lists the conformer; the second and third columns display the RMSD values of the RDKit post-processing results in comparison with MCLMCS results, with and without the addition of hydrogen, respectively; the fourth and fifth columns present the RMSD values of the MacroModel post-processing results, with and without hydrogen comparison with MCLMCS, respectively. 61
2.16 The MM3 energy values in kcal/mol for each conformation of **c5n** found by *Diamond Energy* RDKit post-processing pipeline, *Diamond Energy* MacroModel post-processing pipeline and MCLMCS (kcal/mol). 62
2.17 The MM3 energy values in kcal/mol for each conformation of **c8a** found by *Diamond Energy* RDKit post-processing pipeline, *Diamond Energy* MacroModel post-processing pipeline and MCLMCS (kcal/mol). 62

2.18 The MM3 energy values in kcal/mol for each conformation of **c8s** found by *Diamond Energy* RDKit post-processing pipeline, *Diamond Energy* MacroModel post-processing pipeline and MCLMCS (kcal/mol). 63
2.19 The table presents the lowest energy values calculated using the MM3 method (in kcal/mol), as identified by various techniques. These include *Diamond Energy* conformer generation followed by RDKit hydrogen post-processing and then minimization, *Diamond Energy* conformer generation followed by MacroModel hydrogen post-processing and then minimization, as well as results from the SPMC, MCMM, MCLMCS, and LMCS methods. 72
2.20 The table presents the lowest energy values calculated using the OPLS4 method (in kcal/mol), as identified by various techniques. These include *Diamond Energy* conformer generation followed by RDKit hydrogen post-processing and then minimization, *Diamond Energy* conformer generation followed by MacroModel hydrogen post-processing and then minimization, as well as results from the SPMC, MCMM, MCLMCS, and LMCS methods. 74
2.21 The table displays the minima calculated using the MM3 method, identified through various techniques. This includes conformer generation with *Diamond Energy* followed by RDKit hydrogen post-processing and minimization, conformer generation with *Diamond Energy* followed by MacroModel hydrogen post-processing and minimization, and the results from the SPMC, MCMM, MCLMCS, and LMCS methods. 78
2.22 The table displays the minima calculated using the OPLS4 method, identified through various techniques. This includes conformer generation with *Diamond Energy* followed by RDKit hydrogen post-processing and minimization, conformer generation with *Diamond Energy* followed by MacroModel hydrogen post-processing and minimization, and the results from the SPMC, MCMM, MCLMCS, and LMCS methods. 79
2.23 Conformers within a 3 kcal/mol range of the lowest energy, identified by various pipelines using MM3 calculations, include those generated with *Diamond Energy* followed by RDKit hydrogen post-processing and minimization, *Diamond Energy* followed by MacroModel hydrogen post-processing and minimization, along with results from the SPMC, MCMM, MCLMCS, and LMCS methods. 80

2.24 Conformers within a 3 kcal/mol range of the lowest energy, identified by various pipelines using OPLS4 calculations, include those generated with *Diamond Energy* followed by RDKit hydrogen post-processing and minimization, *Diamond Energy* followed by MacroModel hydrogen post-processing and minimization, along with results from the SPMC, MCMM, MCLMCS, and LMCS methods. 81
2.25 Conformers within a 5 kcal/mol range of the lowest energy, identified by various pipelines using MM3 calculations, include those generated with *Diamond Energy* followed by RDKit hydrogen post-processing and minimization, *Diamond Energy* followed by MacroModel hydrogen post-processing and minimization, along with results from the SPMC, MCMM, MCLMCS, and LMCS methods. 82
2.26 Conformers within a 5 kcal/mol range of the lowest energy, identified by various pipelines using OPLS4 calculations, include those generated with *Diamond Energy* followed by RDKit hydrogen post-processing and minimization, *Diamond Energy* followed by MacroModel hydrogen post-processing and minimization, along with results from the SPMC, MCMM, MCLMCS, and LMCS methods. 83

3.1 Hyperparameters Verifying Dataset Used for Ether Oxygen Related Terms. 112
3.2 Lowest Energy and Boltzmann Factor based Performance Score Comparison between *Diamond Energy* II and MCLMCS for Ether Oxygen Related Energy Terms (OPLS4 kcal/mol). 117
3.3 Hyperparameters Verifying Dataset Used for Alcohol Oxygen-Related Terms. 118
3.4 Lowest Energy and Boltzmann Factor based Performance Score Comparison between *Diamond Energy* II and MCLMCS for Alcohol Oxygen Related Energy Terms (OPLS4 kcal/mol), and the corresponding molecules shown in Table 3.3. 118
3.5 **e4** example - low energy conformers identified of *Diamond Energy* II method and MCLMCS method, using MM3 calculations (kcal/mol). 123

3.6 Comparison of Lowest Energy Values Across Test Molecules Using Six Pipelines. This table presents a side-by-side comparison of the lowest energy values obtained for each test molecule by six different pipelines: Diamond_RDKit, Diamond_MacroModel, MCMM, SPMC, LMCS, and MCLMCS, all utilizing the MM3 Force Field (kcal/mol). The first column lists the names of the test molecules, while the subsequent columns display the lowest energy values identified by each of the six methods. These values are the result of a comprehensive conformer search and minimization process conducted in MacroModel. 125
3.7 This table presents statistics including the mean, standard deviation, minimum, 25th percentile, median (50th percentile), 75th percentile, and maximum values for the 'Lowest_Total_Energy' achieved by each method. Additionally, a column titled 'Lowest Energy Found Times' is included, which indicates the frequency with which each method identified the lowest energy value for a specific molecule in the test set. This table offers a comprehensive view of the central tendency and variability of the energy values obtained by each method using MM3. 126
3.8 Comparison of Lowest Energy Values Across Test Molecules Using Six Pipelines. This table presents a side-by-side comparison of the lowest energy values obtained for each test molecule by six different pipelines: Diamond_RDKit, Diamond_MacroModel, MCMM, SPMC, LMCS, and MCLMCS, all utilizing the OPLS4 Force Field (kcal/mol). The first column lists the names of the test molecules, while the subsequent columns display the lowest energy values identified by each of the six methods. These values are the result of a comprehensive conformer search and minimization process conducted in MacroModel. 129
3.9 This table presents statistics including the mean, standard deviation, minimum, 25th percentile, median (50th percentile), 75th percentile, and maximum values for the 'Lowest_Total_Energy' achieved by each method. Additionally, a column titled 'Lowest Energy Found Times' is included, which indicates the frequency with which each method identified the lowest energy value for a specific molecule in the test set. This table offers a comprehensive view of the central tendency and variability of the energy values obtained by each method using OPLS4. 130
3.10 Conformers Identified Within a 3 kcal/mol Range from the Lowest Energy Using MM3 Force Field Calculations for Six Computational Pipelines. . . . 133

3.11 Average Number of Conformers (Within a 3 kcal/mol Range of the Lowest Energy) Identified by Each Method Using MM3 Force Field Calculations. . 134
3.12 Conformers Identified Within a 5 kcal/mol Range from the Lowest Energy Using MM3 Force Field Calculations for Six Computational Pipelines. . . . 135
3.13 Average Number of Conformers (Within a 5 kcal/mol Range of the Lowest Energy) Identified by Each Method Using MM3 Force Field Calculations. . 136
3.14 Conformers Identified Within a 3 kcal/mol Range from the Lowest Energy Using OPLS4 Force Field Calculations for Six Computational Pipelines. . . 137
3.15 Conformers Identified Within a 5 kcal/mol Range from the Lowest Energy Using OPLS4 Force Field Calculations for Six Computational Pipelines. . . 138
3.16 Average Number of Conformers (Within a 3 kcal/mol Range of the Lowest Energy) Identified by Each Method Using OPLS4 Force Field Calculations. 139
3.17 Average Number of Conformers (Within a 5 kcal/mol Range of the Lowest Energy) Identified by Each Method Using OPLS4 Force Field Calculations. 139
3.18 Total Running Time Cost of Each Method to Generate Conformers for All Test Cases. 140
3.19 Average Running Time Cost of Each Method to Generate Conformers for All Test Cases. 141
3.20 Total Running Time Cost of Each Method to Generate Conformers for All Test Cases Using MM3 Calculations. 144
3.21 Average Running Time Cost of Each Method to Generate Conformers for All Test Cases Using MM3 Calculations. 145
3.22 Total Running Time Cost of Each Method to Generate Conformers for All Test Cases Using OPLS4 Calculations. 146
3.23 Average Running Time Cost of Each Method to Generate Conformers for All Test Cases Using OPLS4 Calculations. 147

4.1 Comparison of Dihedral Angle Distribution Distances Between Various Models and *Diamond Energy* Data. 167
4.2 Statistic analysis of CGCF, GeoMol and ConfVAE closeness results. 168
4.3 Comparative Results of CGCF Trained on Different Datasets Using MMFF94 Calculations (kcal/mol). 169
4.4 Comparative Results of CGCF Trained on Different Datasets and Systematic Search Algorithm *Diamond Energy* Using MM3 Calculations (kcal/mol). . 170
4.5 Comparative Results of CGCF Trained on Different Datasets and Systematic Search Algorithm Diamond Energy Using OPLS4 Calculations (kcal/mol). 171

4.6 The count of conformers within 3 kcal/mol of the lowest energy using various methods with MM3 calculations. 172
4.7 The count of conformers within 5 kcal/mol of the lowest energy using various methods with MM3 calculations. 173
4.8 The count of conformers within 3 kcal/mol of the lowest energy using various methods with OPLS4 calculations. 174
4.9 The count of conformers within 5 kcal/mol of the lowest energy obtained using various methods with OPLS4 calculations. 177
4.10 The Lowest Energy Identified for Test Molecules of GeoMol Trained on Different Datasets Using MMFF94 Calculations. 178
4.11 The Lowest Energy Identified for Test Molecules of GeoMol Trained on Different Datasets Using MM3 Calculations. 179
4.12 The Lowest Energy Identified for Test Molecules of GeoMol Trained on Different Datasets Using OPLS4 Calculations. 179
4.13 Showing the count of conformers within 3 kcal/mol of the lowest energy obtained using various GeoMol methods and *Diamond Energy* method with MM3 calculations. 180
4.14 The count of conformers within 5 kcal/mol of the lowest energy obtained using various GeoMol methods and *Diamond Energy* method with MM3 calculations. 181
4.15 The count of conformers within 3 kcal/mol of the lowest energy obtained using various GeoMol methods and *Diamond Energy* method with OPLS4 calculations. 182
4.16 The count of conformers within 5 kcal/mol of the lowest energy obtained using various GeoMol methods and *Diamond Energy* method with OPLS4 calculations. 183
4.17 Comparative Lowest Energy Identified Results of ConfVAE and *Diamond Energy* Using MM3 Calculations. 184
4.18 Comparative Lowest Energy Identified Results of ConfVAE and *Diamond Energy* Using OPLS4 Calculations. 185
4.19 The count of conformers within 5 kcal/mol of the lowest energy obtained using various ConfVAE method and *Diamond Energy* method with MM3 calculations. 185
4.20 The count of conformers within 5 kcal/mol of the lowest energy obtained using various *ConfVAE* method and *Diamond Energy* method with OPLS4 calculations. 186

4.21 Comparative Results of ConfVAE and Diamond-PRCG Using MM3 Calculations. 188
4.22 The lowest energy values identified by GeoDiff, ConfGF, and Diamond-PRCG minimization using OPLS4 calculations. 190

Nomenclature

Acronyms / Abbreviations

AE Autoencoder

AMBER Assisted Model Building with Energy Refinement

ARPCK Arnoldi Package

GCF Conditional Graph Continuous Flow

CHARMM Chemistry at HARvard Macromolecular Mechanics

CNFs Continuous Normalizing Flows

ConfGF Learning Gradient Fields for Molecular Conformation Generation

ConfVAE A Conditional VAE Model for Conformation Generation

CSD Cambridge Structural Database

CVs Collective Variables

d(CpG) A dinucleotide sequence in DNA consisting of a cytosine nucleotide followed by a guanine nucleotide, connected by a phosphate group.

d(GpC) A dinucleotide sequence in DNA consisting of a guanine nucleotide followed by a cytosine nucleotide, connected by a phosphate group. This sequence is often mentioned in studies of DNA structure, function, and interactions, as the G-C pairing is known for its stability due to three hydrogen bonds, compared to the A-T pairing which has two.

DFS Depth-First Search

DFT Density Functional Theory

DG Distance Geometry

DNA Deoxyribonucleic Acid

ECEPP Empirical Conformational Energy Program for Peptides

ETKDG Extended Topological Polar Surface Area-Based Atom-Type Definition for Drug-Like Compounds

ETM Energy-based Tiling Model

FEP Free-Energy Perturbation

FMNR Full-Matrix Newton-Raphson Minimization

GA Genetic Algorithm

GB Generalized Born model

GeoDiff A Geometric Diffusion Model for Molecular Conformation Generation

GeoMol Torsional Geometric Generation of Molecular 3D Conformer Ensembles

GNNs Graph Neural Networks

GPU Graphics Processing Unit

GraphDG A Generative Model for Molecular Distance Geometry

HF Hartree-Fock

JBW Jump Between Wells

JNK3 c-Jun N-terminal kinase 3

LLMOD Large Low Mode Molecular Dynamics

LMOD Low Mode Molecular Dynamics

MCLMCS Monte Carlo combined with the Low-Mode Search Method

MC Monte Carlo

MCMM Monte Carlo Molecular Modeling

MD Molecular Dynamics

ML Machine Learning

MM2 Molecular Mechanics 2

MM3 Molecular Mechanics 3

MMFF Merck Molecular Force Field

MM Molecular Mechanics

MPNNs Message Passing Neural Networks

NLP Natural Language Processing

NMR Nuclear Magnetic Resonance

N The number of particles

NOE Nuclear Overhauser Effect

OPLS Optimized Potentials for Liquid Simulations

PB Poisson-Boltzmann model

PDB Protein Data Bank

PES Potential Energy Surface

PMF Potential of Mean Force

PRCG Powell-Reeves Conjugate Gradient Minimization

QM9 Quantum Mechanics in Nine Atoms

QM Quantum Mechanics

RDKit Open-Source Cheminformatics Software

SD Stochastic Dynamics

SE(3) Euclidean group in three dimensions, which encompasses all possible rigid-body transformations (translations and rotations) in 3D space.

SPMC Unbounded Systematic Search Method

TI Thermodynamic Integration

T The temperature

US Umbrella Sampling

VAE Variational Autoencoder

V The volume of the system

Chapter 1

Introduction

The conformation of a molecule generally refers to the spatial arrangement of atoms in a molecule.

Molecules can adopt various conformations. For example, cyclohexane can exist in chair, twist-boat, and boat forms. Conformations usually have an impact on the molecule's physical and chemical properties. Thus, to have a comprehensive understanding of a molecule's properties, it is essential to understand all of its low-energy conformations.

Each conformation of a molecule is associated with a specific energy, as defined by a particular energy function. When we map these energies against their corresponding conformations in a coordinate system, we create what is known as the Potential Energy Surface (PES) of the molecule. Low-energy states (conformers), are generally situated in the valleys of this energy landscape. These low-energy conformers are of particular interest because they are more stable. As a result, they are more likely to be prevalent in the molecular system and thus have a greater impact on the system's overall properties.

If the energies of all possible conformations are gathered, a detailed Potential Energy Surface (PES) can be constructed. Understanding the PES enables the quick identification of conformers. However, obtaining an accurate PES is often challenging, making the task of identifying conformers far from trivial. Methods for searching for conformers typically start by generating a wide range of initial conformations to explore the PES as comprehensively as possible. This is followed by energy estimations for these sample points and subsequent optimization processes aimed at finding the low-energy conformers. The energy of these conformations is generally calculated using either quantum mechanics (QM)

or molecular mechanics (MM). Additionally, the effects of solvents need to be considered in these evaluations, as molecules often exist in a solvated state. Detailed discussions about global search techniques for PES, energy calculation and minimization methods, and the influence of solvents will be addressed in subsequent sections.

1.1 Understanding the Crucial Role of Conformation Searching in Molecular Modeling

Conformational analysis refers to the study of molecular conformations and understanding the impact of these conformations on molecular properties. In the earliest studies of conformational analysis, physical handheld models such as the Dreiding models [2]. Corey–Pauling–Koltun (CPK) models [3] and ball-and-stick models [4, 5] were commonly used. These handheld models helped early twentieth-century experimental chemists understand molecular structures and hypothesize about experimental mechanisms. With the development of computer software, these handheld models can also be integrated into graphical interfaces [6–8]. Predefined fragment libraries allow scientists to connect fragments of interest with a graphical interface. Energy calculations and structural optimizations can be performed using the software's energy equations. These early models could not easily investigate the possibility that a given fragment may be able to exist in more than one conformation.

Today's computational molecular modeling programs can accomplish conformational searches in many ways [9–11]. Studies in computational molecular modeling today usually begin by taking a three-dimensional structure of the target molecule as input and a conformational search is then done to locate low-energy conformers. This serves as the foundation for ensuing computational investigations, using either Quantum Mechanics (QM) or Molecular Mechanics (MM) methodologies [9–11].

Molecules often exist in multiple conformations, particularly at room temperature or under physiologically relevant conditions. These conformational states are intrinsically related to the molecule's physical and chemical properties. Therefore, a comprehensive analysis of all low-energy conformations is crucial for a thorough understanding of the molecule's behavior. For instance, in the DP4 series of studies by Smith, Howarth, and Goodman [12–14], the goal of which was to align experimental NMR results with the correct structure of the candidate molecule. In these studies, an initial rigorous conformational search is done first. The

subsequent calculations then incorporate these low-energy conformations in a Boltzmann-weighted manner to produce NMR data that closely match the experimental data. Without the conformation search, the simulated data is often very different to the experimental data. Another example comes from Grimme's research [15]. He found that NMR results based on a complete conformational search are closer to the real NMR spectrum than those derived from a single conformation.

In the study of chemical reactions, examining a wide range of molecular conformations allows for a more nuanced understanding of the system's potential energy surface. This deeper insight, in turn, aids in pinpointing the transition states that link different phases of the reaction. As a result, it provides valuable information for understanding the underlying reaction mechanisms [16–25].

In the realm of pharmaceutical research, a key area of focus is the dynamic behavior of molecular conformations during drug-target interactions. Understanding this behavior allows for the quantification of pivotal parameters. These include the thermodynamic stability of the newly formed drug-target complex and the associated changes in free energy. Such parameters are integral for the effective evaluation of potential drug candidates, particularly when employing computational methods like virtual screening [26].

Given the wide range of uses for conformational searching calculation, developing and improving existing algorithms for conformational analysis is critically important. These computational tools are essential for various types of further analysis, both in academic research and in specialized industries.

1.2 Conformation Searching Algorithms Survey

Manipulating molecular structures usually starts by defining a coordinate system and identifying related variables. There are three common coordinate systems for this work. The first is external coordinates, also known as Cartesian coordinates, which define the position of each atom in terms of x y and z coordinates. The second is internal coordinates. In this conformation searching system, torsion angles are treated as the variables, and bond lengths and angles are usually kept constant [27]. This focus on torsion angles is because they are usually the main differences between different conformers of the same molecule. Changing bond lengths or angles is hard because it requires a lot of energy to perturb the electron clouds that hold atoms together in a bond. On the other hand, rotating around single bonds usually

requires less energy, making it an easier way to change a molecule's shape. Molecules also vibrate. However, these vibrations around equilibrium bond lengths and angles usually don't have enough energy to overcome barriers and lead to a new stable conformation on the PES [27]. The third system is called distance geometry. It uses matrices of the distances between atomic nuclei to represent the relationships between atoms, initially neglecting the requirement that real molecules are constrained to three-dimensional space.

Within these defined coordinate systems, various strategies are applied to transform the respective variables and generate new conformations, which will then be optimized to low energy configurations. These strategies aim to perform a global search on the molecular potential energy surface. Commonly utilized approaches include build-up methods, systematic techniques, random transformations, genetic algorithms, Molecular Dynamics (MD), distance geometry techniques, ellipsoid-based methods, rule-based approaches, and emerging Machine Learning (ML) methodologies.

The subsequent sections will offer a brief introduction to these various methods for global conformer search on PES and for conformer energy minimization algorithms. It should be noted that the literature in these fields is extensive. These sections present a review highlighting key issues in the field, rather than a comprehensive literature survey which would be unmanageably large.

1.2.1 Global Search for Molecular Structures Generation

1.2.1.1 Build-Up Approaches

In the earliest days, the strategy for identifying low-energy conformations of a target molecule involved building up its low-energy fragments together [28–32]. These low-energy fragments were determined through energy minimization calculations [28, 31]. The underlying assumption of this approach was that the behavior of the entire molecule could be approximated by the behavior of its smaller components, implying that short-range interactions dominate the formation of low-energy conformations [27].

In the early 1980s, Scheraga investigated peptides with a method which used a pre-optimized library of single residues with terminal blocks to assemble the Met-enkephalin molecule [28–30]. The initial stage of the process involves forming a dipeptide by selecting only the conformations with the lowest energy levels. These initial combinations are further refined

through energy minimization and the most optimal conformations are chosen for the following assembly stages. The steps are reiterated until the entire molecular structure is constructed. To enhance the efficiency of the build-up, some strategies were implemented [29]. First, only those conformations that significantly differ in their backbone structures are retained for further use, thereby reducing the increasing memory requirements as the process continues. Second, a special energy minimizer designed specifically for the ECEPP force field was used. This tool maintains rigid bond lengths and angles while allowing only torsional variations. If there is an issue with atoms overlapping during the assembly, the minimization process is reinitiated, using an alternative energy function designed to handle these overlap situations and changes to the standard energy function when no atoms overlap [29, 30].

Another application of this strategy is used in DNA structure prediction [31, 32]. Scientists first conducted global searches and minimization for all sixteen types of deoxynucleoside monophosphates. Then, the minimum energy conformations of d(CpG) were combined with those of d(GpC) to provide initial conformations for the energy minimization of single-stranded d(CpGpC). Next, duplex structures for d(CpGpC) · d(GpCpG) were generated by energy-minimizing combinations of the d(CpGpC) minima in a manner that allowed for dinucleotide repeats. The two lowest energy duplex structures calculated found were Z- and B-helices. In addition, the A-form and many novel Watson-Crick base-paired structures were also identified [32]. At that time, conducting a direct global search for DNA duplex structures posed a challenge due to the complex degrees of freedom involved. The build-up strategy provided a useful alternative for predicting DNA structures.

However, it is important to note that while using low-energy fragments can expedite the search process, it does not guarantee that the resulting structure will have the lowest possible overall energy. This is because energy-favorable components might be discarded during the early stages of fragment minimization and filtering, limiting the thoroughness of the global search and potentially missing lower-energy conformations. Additional limitations include increasing computational memory requirements as the build-up process continue. Furthermore, not all low-energy fragments may be readily available for use.

1.2.1.2 Systematic Methods

The systematic search, also referred to as a deterministic search, is a straightforward approach for exploring molecular conformations. The method is based on the principle that distinct molecular conformations arise primarily from variations in the values of torsional angles.

To execute this search, each rotatable bond is systematically rotated through 360° in small increments, thereby generating all possible combinations of torsional angles.

The early use of this method was reported by Still [33], who designed a program called RINGMAKER to generate conformations for macrocycles. The program starts with the ring in an open conformation, and then exhaustively explores all possible combinations of dihedral angles, with an increment of 10 degrees or more. After generating these configurations, the program selects those with reasonable bond distances and angles at the end to successfully close the ring. Similar conformational searches for macrocycles have also been reported, such as the work of Fukazawa [34].

Since closure constraints are typically applied late in the process—once most of the ring system has been formed—rejections at this stage are relatively inefficient. Several studies have focused on understanding the conditions for ring closure, aiming to improve the efficiency of these types of ring-searching algorithms. Smith and Veber [35] worked to enhance these ring searching algorithms. For instance, they introduced selective checks for van der Waals contacts to swiftly discard unfavorable search branches. Additionally, they evaluated whether the remaining section of a molecule, in its most extended form, could close the ring during the second half of the search. Lastly, they attempted to incorporate experimental data, such as dihedral angles and proximity contacts measured by Nuclear Overhauser Effect (NOE), to guide the search toward more accurate structures. Go and Scheraga [36] developed a mathematical method to calculate the set of torsional angles needed to connect two specific points in space. Their research revealed that when dealing with six or fewer dihedral angles, the problem becomes fully constrained, meaning that the torsional angles are not independent but are strictly determined by the endpoints. Bruccoleri and Karplus [37] enhanced their macrocycle conformer generation by building upon the work of Go and Scheraga [36], while also incorporating variations in bond angles as an improvement. They found that this added flexibility is particularly useful for modeling complex structures, such as alpha-helices or those based on low-resolution data.

Other early classic algorithms include MOLTW [38], which employs a recursive algorithm to efficiently explore combinations of torsion angle rotations and simply checks for overlapping atoms. Conformers that pass this check are then subjected to further energy calculations. Additionally, some approaches focus on limiting the search range for using the Ramachandran map for polypeptides [39] or pharmacophore constraint for molecules of interest [40].

Still and coworkers [41] significantly improved systematic search methods by introducing an effective algorithm for tree-searching in internal coordinates. This approach is somewhat akin to the successor of 'RINGMAKER' [33] but it is applicable to molecules beyond just macrocycles. The algorithm begins by selecting one end of the molecular chain and iteratively altering its dihedral angles (torsion angles) as it travels down the chain. For cyclic molecules, the algorithm temporarily 'opens' the ring to treat it as a linear chain, allowing the same logic applied to acyclic molecules to also work for cyclic ones. However, additional constraints are imposed to ensure the ring can later be closed back into its proper cyclic form. To enhance efficiency, the entire process is represented as a tree structure. Each node in the tree corresponds to a specific conformational state of the molecule, and the edges between nodes represent the torsional rotations that transition the molecule from one state to another. This structured approach facilitates easier navigation and processing of the vast number of possible conformations. During the process, early constraint tests are employed to quickly eliminate any molecular conformations that are chemically unreasonable. For instance, the algorithm checks for minimum allowable distances between certain non-bonded atoms to avoid steric clashes. This constraint-based elimination is crucial for expediting the computational process, as it enables the algorithm to 'prune' large branches of the tree leading to invalid structures, thereby saving computational time by preventing chemically unfavored structures from entering into the subsequent minimization step, which is more time-consuming [41]. In addition to introducing this efficient search algorithm, they also explored how to determine the optimal increments for dihedral angles in grid searches. Through their studies on hydrocarbons, they found that a dihedral step size of 60° was sufficient to capture all possible conformations. Karplus and colleagues introduced a tree-searching process similar to the one described, specifically for short polypeptide segments in proteins, within their program named CONGEN [42]. However, to prune branches associated with unfavorable geometries, they employed the empirical energy function provided by CHARMM [43].

Goodman [44] introduced a refinement to Still's tree-searching strategy [41] for conformational analysis. In Goodman's approach [44], the algorithm prioritises conformational diversity in a systematic way, starting with large changes to torsion angles and progressively refining the search to higher resolutions. Unlike previous methodologies, this novel approach allows for the early generation of torsionally diverse conformations. Additionally, the algorithm is unbounded, permitting the search to either continue indefinitely or terminate upon reaching preset conditions, such as the discovery of all new conformers or the attainment of a pre-defined maximum resolution.

The challenge inherent in this type of algorithm is that the computational complexity increases exponentially with the number of rotatable bonds, leading to a combinatorial explosion. As a result, optimizing the search speed becomes a critical focus in systematic search algorithms [45].

Crippen and Scheraga [46] had introduced the concept of 'ravine-following', which they termed the 'Method of Gentlest Ascent'. They posited a procedure in which an algorithm begins by identifying a local minimum in a low-energy region of the PES. From this starting point, the algorithm selects a path along which the energy incrementally ascends, with the aim of passing through a low-lying saddle point, also described as a 'mountain pass'. Subsequent energy minimization then guides the system to the next local minimum. Crippen and Scheraga suggested that this process could be iteratively repeated to explore a sequence of adjacent minima on the PES. While mode-following techniques have been employed for exploring the PES in various contexts [25, 47], their application to conformational search has not been extensively studied. This idea was a possible resolution to Levinthal's paradox [48], which suggests that proteins seem to fold much more quickly than would be possible if they were merely randomly exploring conformation space. It is not clear that this idea is relevant to the conformation analysis of small molecules, however.

Kolossváry and Guida developed a novel mode-following algorithm specifically designed for systematic conformer search. This new method, termed 'Low-Mode Search' (LMOD) [16], shares conceptual similarities with the original Method of Gentlest Ascent and offers a refined approach for traversing the intricate landscape of molecular conformations.

In the Low-Mode Search (LMOD) [16], the procedure starts with an arbitrary local energy minimum. A subsequent normal mode analysis on this initial configuration identifies its low-frequency vibrational modes. These modes are recorded as an array of eigenvectors based on a user-specified frequency threshold. To map out the conformational landscape, LMOD incrementally alters the minimum structure along the vectors indicated by these low-frequency modes. If the potential energy crosses a user-set threshold during these alterations, the algorithm applies energy minimization to that structure to identify a new local minimum. One feature of this method is it does not update the Hessian matrix during this search, which leads to computational efficiency. As the exploration progresses, LMOD accumulates a set of new local minima. These newly identified minima serve as starting points for doing the same investigations along their corresponding low-frequency modes, ensuring a thorough exploration of the entire potential energy surface. The exploration is limited by the number of low-frequency modes identified. When all systematic pathways have been explored for

a local minimum, the algorithm transitions to a Monte Carlo-based strategy. During the stochastic search, it explores directions that are random combinations of the low-frequency eigenvectors, employing the same procedure used in the systematic exploration. The Hessian matrix also does not re-calculate in this search process. LMOD is considered among the most efficient methodologies for conformational search in current use.

Later, an extension algorithm called large LMOD (LLMOD) [17], was developed by Kolossváry for large-scale conformational searching. It aims to solve fully flexible, unconstrained protein structures for complex loop optimization problems. This algorithm is particularly tailored for handling complex loop optimization problems in fully flexible, unconstrained protein structures. Unlike the original LMOD, which requires the explicit calculation and storage of the Hessian matrix—a task unfeasible for structures exceeding 2000 atoms—LLMOD employs ARPACK software to efficiently compute low-mode eigenvectors using an implied Hessian matrix. This calculation is based on gradient methods, simplifying the computational process. Moreover, previous analyses have shown that low-frequency vibrational modes exhibit similar collective motions across various protein conformations. This observation implies that these low-mode eigenvectors are transferable between different structures, allowing LLMOD to avoid recalculating eigenvectors for each new conformation identified. This optimization further enhances the algorithm's computational efficiency. The efficacy of LLMOD has been validated through tests on a complex protein model, specifically the c-Jun N-terminal kinase JNK3.

Inspired by the success of LMOD/LLMOD, lots of similar mode-following style conformer search methods were developed, though not necessarily systematic strategies. For example, researchers have formulated a new molecular dynamics technique called LowModeMD [49], designed to focus on low-frequency atomic movements applied to molecular dynamics. This approach utilizes a stochastic framework to execute brief molecular dynamics simulations. The process starts by generating random atomic velocities. These are then filtered through a Chebyshev polynomial filter to emphasize low-frequency vibrational modes, without the use of either an implicit or explicit Hessian. Distinct from LMOD/LLMOD, LowModeMD avoids the need for computationally intensive calculations such as eigenvectors and second derivatives, rendering it advantageous for larger and more complex molecular structures. Through iterative application of these steps, LowModeMD effectively yields a diverse set of low-energy conformations, thereby proving suitable for the stable configuration analysis of intricate molecular systems. In one study comparing the efficiency of Monte Carlo and LMOD methods [50], it was found for larger molecular systems, a balanced 50:50 combination of Low Mode and Monte Carlo methods proved to be the most efficient. Also,

to avoid repeatedly sampling areas already explored on the Potential Energy Surface (PES), some methods use a memory feature. This improves the efficiency of mode-following methods [51, 52]. Owing to their computational efficiency, these mode-following methods have garnered extensive application across diverse research areas, notably in the domains of protein perturbation and molecular locking analysis [47, 53–56].

1.2.1.3 Random Transformations

To mitigate the computational demands associated with grid search techniques, stochastic algorithms have been increasingly researched and developed. One of the most prominent in this category is the Monte Carlo method. Monte Carlo consists of a range of computational algorithms that employ random sampling to approximate solutions.

In a Monte Carlo-based conformer search, the process typically begins by introducing a random alteration to the molecule, either through torsion rotation or changes in Cartesian coordinates. The energy of the resulting geometry is then calculated and compared with the previous configuration. If the new geometry has lower energy, it is accepted as a new conformer. If the energy is higher, researchers often employ decision rules to determine whether to accept the new geometry. The Metropolis criterion [57] is commonly used for this purpose: the probability of accepting a higher-energy conformer is calculated based on the Boltzmann distribution and is given by $P = \exp(-\Delta E/kT)$, and it is usually terminated when no new low-energy structures are found. Utilizing the Metropolis criterion enables the algorithm to avoid becoming trapped in local energy minima and allows it to sample the conformational space in accordance with the Boltzmann distribution.

Early implementations of Monte Carlo methods have been identified as less efficient than alternative approaches in protein conformer generation [58]. This inefficiency might be attributed to the sampling of molecular structures, like proteins, where steep energy surfaces are predominant, resulting in a low overall acceptance rate for new configurations [58]. An enhancement to the Monte Carlo technique has addressed this issue [59]. In this improved version, bond lengths and angles remain fixed while random adjustments are made exclusively to the dihedral angles. Moreover, the challenge of anisotropic energy surfaces affecting acceptance rates has been managed by employing smaller step sizes in steeper regions and larger steps in flatter areas.

Scheraga [60] presented an approach that utilizes the Metropolis Monte Carlo algorithm for investigating protein folding. Within this framework, a randomly chosen torsion angle in the molecule is modified by a random angular increment. The molecular structure then undergoes an energy minimization process for refinement. Following this, the Metropolis criterion [57] is applied to determine whether the newly refined structure should be retained or discarded, thereby setting the stage for the next iteration of random adjustments to torsion angles. In their analysis, they found that a dozen different starting structures converged to the same energy minimum when tested in a vacuum using this method. However, under aqueous conditions, they were unable to identify a stable configuration. This suggests that multiple structures coexist in a state of equilibrium when in water.

Subsequently, Still [61] introduced a method termed 'Monte Carlo Multiple-Minimum (MCMM) Search'. In this approach, an initial molecular configuration is chosen, and stochastic modifications are applied to internal coordinates. The modified structure then undergoes an energy minimization process, after which the resulting conformation is compared to those identified in prior search iterations. If the configuration is unique and within the energy threshold compared to the instant global minimum, it is saved as a new conformer; otherwise, it is rejected. Each iteration of this process is labeled as an MC step. The study reveals that combining the search process with previously identified low-energy conformers enhances efficiency when compared with a random walk approach. Therefore, a key determinant of search efficiency is the algorithm's approach to selecting low-energy starting geometries for each MC step, which will help guide the search bias to low-energy areas.

Saunders [62, 63] introduced a stochastic search algorithm based on Cartesian coordinates. The approach initiates from a structure close to a local energy minimum. A random perturbation, commonly referred to as a 'kick', is then applied to this initial structure. A random number generator calculates an independent random change for each atom's coordinates, within a range of 1.5 Å to 3 Å. After this perturbation, the modified structure undergoes an optimization process to find a new local energy minimum. Initially, an in-house fast two-body force field [64] is used to quickly eliminate structures distorted by the random kick, to expedite the minimization process. Subsequently, the remaining reasonable structures are optimized using the MM2 force field [65]. The algorithm yields one of two outcomes: it either returns to the initial conformer or arrives at a new one. To distinguish new conformations from previous ones, the method keeps a record of all unique minimum-energy configurations encountered.

A random method based on Cartesian coordinates has been reported by Raber [66]. They used a coordinate truncation strategy to generate structures and then minimized the structures using MM2 force field [65]. The truncation procedure consists of altering the output geometry of a structure that has been minimized. Specifically, the x, y, and z coordinates for each atom's deviations are truncated to either two or three decimal places, thereby introducing minor perturbations in atomic positions. This perturbation shifts the structure to a different point on the potential energy surface, which is no longer situated at a local energy minimum. Subsequent analytical minimization provides an opportunity for a search directed toward finding a new minimum-energy configuration.

In scenarios where high energy barriers isolate low-energy regions and restrict exploration of the full potential energy landscape, standard Monte Carlo methods (Metropolis Monte Carlo) tend to be inefficient. To circumvent these constraints, Still and colleagues [67, 68] introduced a hybrid method called Jump Between Wells Monte Carlo-Stochastic Dynamics (JBW/MC-SD). Unlike the standard Monte Carlo method, which employs random changes in each searching step, this new approach leverages prior knowledge of low-energy conformers to guide the exploration of the potential energy surface (PES). Initially, a comprehensive search [44] identifies all possible low-energy conformers, which are stored in a set. The transformations needed to switch between any two conformers in this set are also recorded. These transformations serve as predefined pathways that inform the subsequent steps in navigating between conformers on the PES. Stochastic dynamics are then utilized to further explore each identified energy well.

A method combining Monte Carlo simulation with simulated annealing was introduced [69], leveraging the Metropolis algorithm to simulate molecular conformations at designated temperatures. The program employs a strategically decreasing temperature sequence to exploit the temperature-dependent nature of the Boltzmann distribution. As the temperature is systematically reduced, molecular conformations are effectively 'frozen out'. Ultimately, the molecule becomes confined to a specific energy well. When the 'cooling schedule' is optimally configured, this process ensures that the molecule attains its global minimum energy conformation.

1.2.1.4 Genetic Algorithm (GA)

The Genetic Algorithm (GA) is a calculation method inspired by the process of natural selection and genetics. It is first introduced by John Holland, an American computer scientist

in the 1960s [70], and it mimics the natural evolution process by representing the problem to be solved as strings of binary values. The fitness of the current state is assessed, then the appropriate parents are selected for crossover and mutation. This process is repeated until satisfactory solutions are identified. It is widely used in various fields as a powerful tool to help escape local minima traps in global optimization problems. Conformational searching is essentially a global optimization problem whereby the objective of a conformational search is to optimize the potential energy function of a molecule setting the variables so that the coordinates define the molecular conformation. Applying GA in conformational searching should be able to help to explore PES effectively.

A classical genetic algorithm example has been developed by Goodman and coworkers [71], in which the study presents a GA designed to optimize the conformations of unbranched alkanes by focusing on dihedral angles as key variables. Initially, the algorithm generates a pool of 10 random conformations, each represented as a "chromosome" (strings of binary values) of real numbers corresponding to dihedral angles. The energy of each conformation is minimized to assess its "fitness." Subsequent generations are created through crossover and mutation operations, selectively biased towards lower-energy states. Both parent and new conformations are considered for progressing to the next generation. The algorithm also incorporates measures to maintain diversity in the pool. After running for the required number of generations, the algorithm produces a list of all encountered conformations, sorted by their energy levels.

Other genetic algorithm based searches have been reported in the literature, and they follow a similar strategic framework [72–77]. The primary variations lie in the choice of string representation (gray string, binary string etc.), the frequency of crossover events, the mutation rate, and other intermediary parameters.

1.2.1.5 Molecular Dynamics (MD)

Molecular dynamics simulations employ Newton's equations of motion to calculate the movement of individual atoms under the influence of both intramolecular and intermolecular forces such as Lennard-Jones and electrostatic interactions, thereby determining the trajectories of molecules. This computational approach allows for the observation of dynamic changes in complex systems. A significant advantage of molecular dynamics over other conformer search methods is its ability to compute time-dependent properties [11, 78, 79].

The calculation of particle motion within a continuous potential field is inherently complex. To integrate the equations of motion in molecular dynamic, two primary numerical methods are commonly employed: the Finite Difference Method and the Predictor-Corrector Integration Method [11]. In the Finite Difference Method, the time integration is divided into discrete intervals, each denoted by a fixed time step Δt. During each time step, forces acting on the particles are assumed to be constant. Algorithms based on this method, such as the Verlet algorithm, calculate the trajectories of atoms by considering initial conditions and forces acting on them. Common variants of this approach include the Leapfrog algorithm, the Velocity Verlet method, and Beeman's algorithm [11]. In Predictor-Corrector Integration Methods, there are three key steps. The first involves using Taylor series expansions to make initial predictions for new particle positions, velocities, and accelerations. The second step involves calculating the forces at these predicted positions to obtain updated accelerations, denoted as $a(t+\Delta t)$. These updated accelerations are then compared with the initial predictions. The final step refines these initial predictions by correcting the positions and velocities based on the differences between the updated and initially predicted accelerations [11].

Integration methods could be used for computing the trajectories of molecular dynamics simulations, they can be computationally intensive. Several strategies have been developed to improve the computational efficiency. One such method is Multiple Time Step Dynamics [11]. The underlying principle is to differentiate between forces that vary quickly, such as bonded forces, and those that change more slowly, like van der Waals interactions or distant electrostatic interactions. Forces that change rapidly are updated at each time step, as they have a significant impact on the system's short-term behavior. In contrast, forces that vary more gradually are updated less frequently, perhaps every N time steps, to reduce the computational burden. Another commonly utilized optimization technique is the SHAKE algorithm [80]. This algorithm enhances the efficiency of molecular dynamics simulations by allowing for longer time steps without compromising accuracy. Specifically, the SHAKE algorithm maintains the bond lengths at their equilibrium values. After each simulation step in which no constraint is used, the algorithm iteratively adjusts the atomic positions to ensure that the bond lengths remain constant, fine-tuning these positions until they meet a predetermined tolerance level. By constraining these bond lengths, the simulation is able to employ larger time steps when solving the equations of motion.

Some molecular dynamics models have been developed to investigate particular scenarios, such as the constant-temperature ensemble, NVT, Number of particles (N), Volume (V), and Temperature (T) are held constant [11, 81]. It could be used to achieve a stable state at the beginning of the simulation, and maintaining a constant temperature can facilitate this. Also,

studying phenomena like protein unfolding or glass formation often necessitates control over temperature. Temperature regulation in molecular dynamics is typically achieved through two primary methods. The first approach is Velocity Scaling, wherein the kinetic energy of the system is proportional to its temperature. By scaling the velocities of individual particles, the temperature can be adjusted. A straightforward formula is utilized at each time step to carry out this scaling. The second method is Heat Bath Coupling, which involves linking the system to an external 'heat bath' maintained at the desired temperature. The system's temperature is then adjusted based on the temperature differential between the system and the heat bath. This method offers another avenue for precise temperature control in the simulation [11, 81].

Another model of interest is the Constant Pressure model [11], which could be implemented for enhancing the experimental relevance of the simulation. Many experimental conditions are controlled at constant temperature and pressure, making simulations under these same conditions more directly comparable to experimental outcomes. Also, this model enables the study of intricate phenomena, such as pressure-induced phase transitions. To maintain a constant pressure within the simulation, three common approaches are employed. The first approach involves volume adjustments, where the simulation alters its volume dynamically to keep the pressure constant. The second is Pressure Bath Coupling. The system is coupled to an external 'pressure bath', operating analogously to the heat bath used for temperature control. The third approach employs an Extended System for Pressure, introducing an additional degree of freedom that mimics a virtual 'piston'. This piston possesses both kinetic and potential energies and can interact with the system to modulate the volume, and consequently, maintain constant pressure [11].

Sometimes, solvent effects are crucial when simulating solute behavior. However, directly simulating every solvent molecule is computationally expensive. To address this, the Potential of Mean Force (PMF) is frequently utilized. PMF quantifies the average influence exerted by the solvent on the solute, which is represented in terms of changes in free energy. This approach allows the simulation to replicate solvent effects without explicit modeling of individual solvent molecules. Another commonly employed method is Stochastic Dynamics. This involves the inclusion of a stochastic term in the equations of motion, accounting for force fluctuations experienced by individual atoms. This stochastic term allows for the representation of complex or disordered environments like solvents. These random forces are generally modeled to follow a Gaussian distribution [11].

The methodologies discussed thus far predominantly fall under the category of atomistic simulations. While these provide comprehensive details at the atomic level, they come at the cost of high computational expense and are generally restricted to nanosecond timescales. Alternative approaches like Dissipative Particle Dynamics and Coarse-Grained Kinetics offer a simplified view of complex dynamical processes. These methods facilitate simulations over extended time periods and permit the study of larger system sizes [11].

In addition to the classical dynamics methods previously discussed, enhanced molecular dynamics simulation techniques have been increasingly utilized in recent years, particularly in the field of drug development [26, 82]. One notable example is Umbrella Sampling (US), a method designed for the efficient sampling of rare events. These rare events are transitions or changes within a system that occur infrequently within the timescale of standard molecular dynamics simulations, which are usually confined to nanoseconds or microseconds. Such events can include protein folding or significant conformational changes in large biomolecules [82]. In the Umbrella Sampling method, the reaction coordinate is segmented into discrete 'windows'. Each of these windows is simulated independently, with a biasing potential applied to constrain the reaction variables within that window. This biasing potential functions as an 'umbrella', holding the system in a specific region and thereby facilitating exhaustive sampling within that window. Finally, mathematical techniques are employed to remove the bias, often by weighting the contributions from each window based on the applied bias and overlap with adjacent windows. By collecting all the unbiased information, a comprehensive understanding of the entire sampling space is achieved [82].

Thermodynamic Integration (TI) is another analytical tool frequently employed to investigate phenomena such as phase transitions, molecular binding, and conformational alterations in biomolecules [26, 83]. In this method, a continuous pathway is established between two distinct states, denoted as State A and State B, by varying a parameter known as λ. Rigorous sampling at each increment of λ is imperative for achieving precise and reliable results. The method allows for the calculation of the free energy difference between the two states by integrating the collected data across the values of λ. [26, 83] Another technique closely related to Thermodynamic Integration is Free-Energy Perturbation (FEP) methods [26, 84]. In FEP, the transition from State A to State B is not achieved through a continuous alteration of a parameter λ, but rather through a series of discrete simulation 'windows', each characterized by distinct λ values. Within each of these windows, the change in free energy due to a minor modification in λ is calculated. By aggregating these incremental free energy changes across the entire spectrum of λ values, one can estimate the overall free energy difference between State A and State B [26, 84].

In traditional molecular dynamics simulations, systems often become confined to local energy minima, inhibiting the exploration of transitions between different states. Metadynamics [26, 85] addresses this challenge by introducing a history-dependent biasing potential to collective variables (CVs). This biasing potential acts as a deterrent for the system against revisiting areas of the CV space it has already explored, thus facilitating the discovery of new regions and enabling the system to escape local minima. Subsequently, this biasing potential is employed to reconstruct the free energy landscape based on the collective variables, often utilizing reweighting techniques.

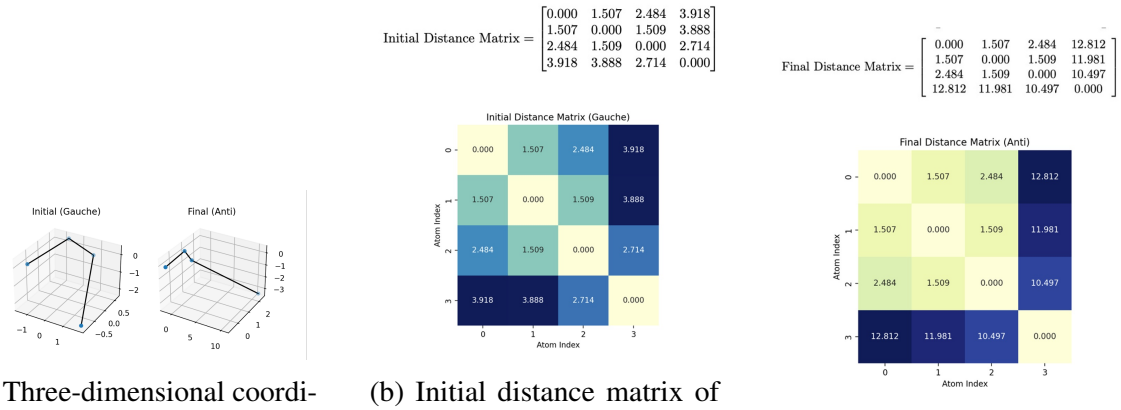
1.2.1.6 Distance Geometry (DG)

In Distance Geometry (DG), the molecular structure is represented by a pairwise distance matrix. This approach seeks to identify a molecule's conformer by optimizing the arrangement of atoms to fulfill a predefined set of pairwise distance constraints. These constraints can either be derived from experimental data, such as Nuclear Magnetic Resonance (NMR) spectroscopy, or be theoretically established based on known bond lengths and angles. In distance geometry, the objective is to find configurations of atoms that satisfies all given distance constraints [86].

DG offers a robust approach to predicting the three-dimensional configurations of molecules, especially when distance constraints between specific atoms are already known from experimental data. As a testament to its effectiveness, the technique has been integrated into the RDKit [87–89] cheminformatics software package. However, the Distance Geometry (DG) method, traditionally employed for this purpose, has limitations: it can yield distorted configurations of aromatic rings and sp2 centers. These distortions often necessitate an additional force field minimization refinement process. Riniker and colleagues [90] introduced an alternative method, known as ETKDG. This new approach enhances the accuracy of DG by incorporating empirically derived torsional-angle preferences and additional rule sets from existing databases. Comparative studies indicate that ETKDG is more successful than traditional DG in accurately reproducing experimentally observed crystal structures.

To clarify the abstract nature of DG, an example involving the generation of butane conformers using RDKit's built-in DG algorithm (ETKDG) [87, 90] is provided to elucidate how DG functions. The computation begins with an input of gauche-form butane, which RDKit [87, 90] automatically transforms into a distance matrix for the molecule. Subsequently, the built-in ETKDG algorithm within RDKit [87, 90] is employed for optimization,

resulting in the generation of the anti-form of butane. The sequence of these transformations is depicted in Figure 1.1a, Figure 1.1b and Figure 1.1c



(a) Three-dimensional coordinates of the conformer generated using the RDKit ETKDG method.

(b) Initial distance matrix of the input molecule as generated by the RDKit ETKDG method.

(c) Final distance matrix of the input molecule as generated by the RDKit ETKDG method.

Figure 1.1 The sequence of transformations for butane.

1.2.1.7 Rule-Based and Knowledge-Based Strategies

Rule-based or knowledge-based methods commonly utilize experimental data, often in the form of crystal structures from databases such as the Cambridge Structural Database (CSD) [91] or the Protein Data Bank (PDB) [92], to inform the generation of molecular conformers. In these approaches, the molecules are disassembled into smaller constituent fragments, and these fragments are then reconstructed using predefined libraries of torsional angles and ring conformations. The reassembly process may occur in either a systematic or stochastic manner, and relies on libraries that contain known conformations of similar fragments. These strategies are seen in programs like Corina [93], OMEGA [94], Confab [95], ConfGen [96], CAESAR [97], and FROG2 [98].

In earlier years, rule-based or knowledge-based approaches also offered efficient constraint conditions to guide the generation of conformers. For instance, Houk and colleagues [99] employed the ellipsoid algorithm [100] to generate conformers, subsequently applying a range of constraints to guarantee the physical plausibility of the generated structures. The ellipsoid algorithm [100] is a well-known optimization technique primarily used for solving convex programming issues. Initially, the algorithm encases a feasible set of solutions in a mathematically defined ellipsoid. During each iterative step, a test point at the center

of the existing ellipsoid is evaluated for its feasibility in relation to specific problem constraints. Based on this evaluation, a new, reduced ellipsoid is constructed, which continues to encompass all potentially feasible solutions. The algorithm systematically narrows down this ellipsoid, focusing it around the most optimal solution. In the context of conformer generation [99, 101], the torsional angle is designated as the system's variable parameter. The ellipsoid algorithm is employed to iteratively reduce the volume of an ellipsoid containing a set of points that represent possible conformations. Various constraints, such as van der Waals forces and specific interatomic distances, are implemented to eliminate unrealistic or infeasible conformations.

1.2.1.8 Emerging Machine Learning (ML) Methods

Dolata and Leach [102] developed one of the earliest AI systems, called "Wizard," for generating molecular conformations. This system uses AI and expert system techniques to systematically explore different shapes that a molecule can take. It breaks down the molecule into smaller parts, assigns known shapes to these parts, and then combines them to form possible overall shapes for the entire molecule. The system then checks these suggestions for any issues or inconsistencies and resolves them. These generated shapes can either be used as they are or further refined using numerical methods.

Utilizing data-driven machine learning models to directly predict molecules and their corresponding low-energy conformations presents challenges. These primarily due to the limited capacity of such models to capture the complex distribution of possible molecular shapes, as well as the intrinsic difficulty of modeling long-range dependencies between atoms. As a result, various machine learning approaches aim to learn the generation of the most stable conformer for a given molecule. Some methods utilize Bayesian optimization [103, 104] to find the most probable or energetically favorable conformer. These methods often use a predefined torsion angle library, based on the existing database, as a training set. This guides the model toward conformations closely resembling experimentally observed crystal structures. Others employ conditional deep generative graph neural networks to predict the most stable conformer [105]. A novel approach is the GeoDiff [106], inspired by the diffusion process observed in classical non-equilibrium thermodynamics. In this framework, atoms are treated as particles that diffuse from their original states to a noisy distribution. The model is trained to reverse this diffusion process, effectively recovering stable molecular conformation from a noise-affected state.

While directly learning the distribution of a molecule's conformations is a challenge, there have been attempts to tackle this issue. One such effort is by Simm et al. [107], who introduced a probabilistic model that employs graph neural networks to learn the conformational distributions of molecules. They utilized the ISO17 dataset [108] for training, which comprises various molecular conformations for every molecule. In their methodology, a molecule is converted into an extended molecular graph, where nodes represent atoms and edges signify molecular bonds or auxiliary connections. From this graph, the conformational distance matrixes are extracted and learned by the model. Once well-trained, the model can generate conformational distance matrices for an input molecule, which can then be used to construct actual 3D conformers. Building upon this work, Tang and colleagues [109] introduced improvements to better capture the multi-modal distribution of conformations. They employed Conditional Graph Continuous Flows instead of graph neural networks, enhancing the model's ability to estimate complex conformational distributions. Additionally, they integrated an Energy-based Tiling Model (ETM) to more explicitly capture the long-range dependencies between atoms. Previous approaches required two steps to generate molecular conformations: initially learning the conformational distance matrix distribution of a molecule, followed by constructing structures based on distance geometry. To streamline this process, two end-to-end generative models have been developed. One of these is GeoMol [110], which utilizes an SE(3)-invariant graph to represent a molecular structure, and the model can effectively learn and extract valuable distribution information from the data. After the training, the model begins by calculating the atom embeddings from a given molecular graph using Message Passing Neural Networks (MPNNs). It then predicts the local 3D structures of individual, non-terminal atoms while considering geometric and chiral constraints. For each bond that connects non-terminal atoms, the model generates a unique torsion angle. Another model, ConfVAE [111], also operates on an end-to-end framework. To capture the rotational and translational invariance of molecular conformations, it still uses pairwise distances among atoms as representatives. Unlike earlier approaches that minimize prediction errors in distance space, ConfVAE employs bilevel programming. This strategy allows for the concurrent optimization of both the problem of predicting inter-atomic distances and the problem of constructing molecular conformations based on those distances.

In addition to the aforementioned strategies, researchers have introduced a novel approach called ConfGF [112]. Getting inspiration from traditional force field methods employed in molecular dynamics simulations, this innovative technique aims to identify stable molecular conformers in a dynamic way. Rather than guiding atomic movement with actual force, ConfGF employs simulated gradient fields related to the log density of atomic coordinates.

The model uses these artificial gradient fields to transition interatomic distances to a stable state.

An alternative approach focuses on learning the energy function of molecular systems. In this context, machine learning models have been trained using quantum physics-based data to forecast energy states within molecules [113–116]. These machine-learned energy models can subsequently be integrated with third-party software to do molecular simulations at the quantum mechanics level, enhancing the accuracy and computational efficiency [113–116].

1.2.2 Local Minimization for Structures Fine Tuning

After a global search to find structures that are close to low-energy minima, these structures are typically further evaluated using MM or QM for energy calculations. Within the energy function being used (where the molecular structure coordinates are the variables), mathematical optimization is typically carried out. The processes are usually 'downhill' methods, where the structure of the molecule is modified in the direction that lowers the value of the energy function according to its gradient.

1.2.2.1 Empirical Force field – Molecular Mechanics (MM)

Molecular mechanics (MM) ignores the electronic motions and calculates the potential energy of a system as a function of the nuclear positions only [11]. These equations model interactions between atoms, such as bond stretching, bond angle bending, torsional forces, electrostatic effects, and non-bonded interactions and so on [11]. Empirical force field methods are generally less computationally demanding, making them suitable for simulating larger systems or for applications where high computational speed is necessary.

Common force fields used in molecular modeling include MM2 [65], MM3 [117], AMBER [118], CHARMM [119], MMFF [120], and OPLS [121] etc. MM2 and MM3 are more focused on small organic molecules, AMBER and CHARMM are tailored for biological macromolecules. MMFF aims to be broadly applicable, and OPLS is optimized for liquid-phase simulations and has variants suitable for both small molecules and macromolecules.

1.2.2.2 Quantum Mechanics (QM)

Quantum Mechanics (QM) provides the foundation for understanding a wide range of phenomena at the atomic level. Classical physics, which includes Newtonian mechanics and electromagnetism, is very successful at describing the behavior of macroscopic objects, such as cars. However, at very small scales, classical physics fails to accurately describe the behavior of particles. QM emerged in the early 20th century to address these limitations [11].

The Schrödinger equation is a foundational equation in QM that describes how the quantum state of a physical system changes with time [11]. It is commonly used to calculate the wave function of a system, which in turn can give information about various properties of the system. However, solving the Schrödinger equation exactly becomes computationally intractable for complex systems with many interacting particles (like most molecules, and materials). Even for systems as simple as molecules with just a few atoms, approximations are often necessary [11].

Consequently, various approximate methods have been developed. One approach is the Hartree-Fock (HF) ab initio method [11]. This method approximates the electronic wave function of a multi-electron system using a Slater determinant formed from single-electron wave functions. It accounts for electron-electron repulsion but simplifies it as an averaged mean field [11]. Additionally, there are semi-empirical methods that utilize experimental data to reduce the computational burden of quantum calculations. While these methods are less accurate, they are significantly faster and are well-suited for very large systems. For systems containing numerous atoms, a hybrid QM/MM approach can be employed. This approach applies quantum mechanics to a specific part of the system (often an enzyme's active site, for instance) while using classical force fields for the remainder. This hybrid method can improve computational efficiency while preserving accuracy in key areas. Density Functional Theory (DFT) is another widely used method for studying the electronic structures of atoms, molecules, and solids. Unlike wave function-based methods, DFT focuses on electron density, offering computational efficiency and often achieving comparable accuracy [11].

1.2.2.3 Solvent Models

Since many chemical and biological processes occur in some sort of solvent (like water, organic solvents, etc.), it is important to consider solvent effects when predicting molecular properties and behaviors. However, explicitly including all solvent molecules in the compu-

tational model would be extremely computationally demanding, so various approximations are used.

In implicit solvent models, the solvent is not represented by individual molecules. Instead, its effects are incorporated into calculations as a continuous medium that surrounds the solute [11]. The Poisson-Boltzmann (PB) model is one commonly used implicit solvent model. This electrostatic-based approach describes the solvent as a dielectric continuum and is often employed to model the behavior of charged particles in aqueous solutions [11]. The Generalized Born (GB) model serves as a simplified version of the Poisson-Boltzmann model and is computationally less demanding. Like the PB model, it also relies on a continuum representation of the solvent [11].

1.2.2.4 Energy Minimized Algorithms

The primary objective of energy minimization is to locate a local minimum on the potential energy surface (PES). The potential energy function of the system, usually based on quantum mechanics or molecular mechanics force fields, is essentially a mathematical equation with molecular coordinates as variables. These minimization algorithms typically begin with an initial structure and iteratively adjust the positions of atoms in the direction that lowers the system's energy. This adjustment follows the energy gradient, which is calculated mathematically. The ultimate goal is to stabilize the molecular geometry at a local minimum on the PES.

Common energy minimization algorithms include the steepest descent method [11], which is the simplest approach wherein the system moves in the direction of the steepest energy gradient. Despite its simplicity, this method can be slow and inefficient. The second algorithm is the conjugate gradient method [11], a more advanced version of the steepest descent algorithm. It utilizes information from previous iterations to predict subsequent steps, making it more efficient for systems with numerous variables. Lastly, there is the Newton-Raphson method [11], which employs not only the first derivative (the gradient) but also the second derivative (the Hessian matrix) to predict the direction of movement. While this approach allows for more thorough calculations, computing the Hessian matrix is computationally demanding.

1.2.3 The Evaluation of a Conformational Searching

Evaluating the effectiveness of conformational searching algorithms presents significant challenges, primarily due to the lack of a definitive benchmark for comparison. The complexity of molecular systems and the vastness of potential conformational spaces contribute to this difficulty, making it challenging to ascertain the accuracy and efficiency of different algorithms.

Commonly adopted methods for conformer search algorithm assessment involve two principal approaches. The first approach involves comparison with experimental crystal data [122]. Many researchers utilize libraries of experimental crystal data as a benchmark to evaluate the performance of their algorithms. The underlying premise is to assess how closely the algorithm's predicted structures align with the observed crystallographic configurations. However, this method has its limitations, as crystal data may not always represent the most energetically favorable conformations in solution due to crystal packing effects. Furthermore, the availability of relevant crystal structures is limited, potentially biasing the assessment towards well-studied molecules.

Another evaluation strategy involves performing a comprehensive search using multiple algorithms, all operating under the same energy model [123]. This comparative analysis allows for a side-by-side assessment of how effectively each algorithm identifies low-energy conformations. While this method offers insights into the relative performance of different algorithms, it does not provide an absolute measure of their accuracy. The reliance on the same energy model also introduces a potential source of bias, as the model's assumptions and limitations could affect all algorithms uniformly, obscuring true differences in their search capabilities.

In conclusion, while the evaluation of conformational searching algorithms is fraught with challenges, it remains a critical area of research. Expanding the toolbox of evaluation methods and exploring new methodologies are essential steps towards improving our ability to accurately predict molecular conformations and, ultimately, advance our understanding of molecular dynamics and interactions.

1.3 Conclusions

It is challenging to rigorously define whether a conformational search algorithm has found the optimal solution, because exhaustive searches of conformation space a so computationally demanding. However, over the past few decades, numerous algorithms have been developed that yield helpful conformer ensemble results. These algorithms have been widely applied in computational molecular modeling.

For complex systems such as proteins, peptides, and nucleic acids, molecular dynamics simulations are the most commonly employed research method. This preference exists for two primary reasons. First, other algorithms often require excessive computational resources to model these complex systems. They are so complex that exhaustive conformation searching is not possible and will not become tractable even if available computer power were to increase by many orders of magnitude. Second, molecular dynamics simulations enable the study of dynamic processes within these systems, allowing for the calculation of time-dependent properties such as diffusion coefficients and changes in the free energy of ligand-receptor binding. Molecular dynamics simulations are currently among the most frequently used methods in pharmaceutical research. However, the limitations in the accuracy of these methods mean that they can only simulate relatively short time scales. Considerable efforts have been made to enhance the precision of molecular dynamics simulations and extend the duration of simulations, with advancements being sought both algorithmically and in hardware capabilities.

For systems that are small enough to do rigorous analyses, systematic search methods are superior to other approaches for generating conformer ensembles [123]. Systematic search can provide reliable results, and given enough time, its search efficiency surpasses that of alternative methods. One area for improvement in systematic search is the computational efficiency for complex molecules. In comparison, random methods are generally less efficient over extended search periods because they may revisit previously explored regions of the conformational space [123]. However, if the goal is to quickly obtain low-energy conformations that adhere to the Boltzmann distribution without considering the full ensemble, random methods can be a viable choice.

Although distance geometry (DG) may be less effective than other algorithms in terms of global search capabilities [123], it excels at efficiently generating conformations with distance constraints. For molecules where experimental distance constraints already exist, distance geometry represents an excellent choice.

Strategies based on build-up, rule-based, and knowledge-based methods can relatively accurately generate low-energy conformations that closely resemble experimental data. However, these approaches are unable to provide comprehensive information about the global potential energy surface (PES). Additionally, it is important to note that molecules may not necessarily exist in the conformations found in crystal structures, as one must also consider the forces involved in the crystal packing.

Emerging machine learning methods can rapidly learn patterns of low-energy conformations from existing conformer databases. However, these methods are database-dependent and may not necessarily provide reliable conformer search results when changing to a different theory level. Additionally, the dataset used might not contain comprehensive information on the distribution of conformers under the corresponding theory level. As high-quality data continue to accumulate, these statistical models may offer greater potential in the future.

Chapter 2

Diamond Energy Method

2.1 *Diamond Energy* Introduction

The *Diamond Energy* approach is based on the observation that low-energy conformers are often found to share a similar framework which resembles a diamond lattice. Whilst this is not true for all molecules, it is often a helpful approximation that should help to survey conformation space quickly. The *Diamond Energy* approach systematically explores molecular conformation space by fitting them into a diamond lattice(Figure 2.1). This has the potential to be rapid because the cartesian coordinates of the atoms in a diamond lattice can be accurately expressed as integers rather than floating point numbers and so the complexity of the calculations is substantially reduced. Local minimization is not required, because all atoms are placed on one of these integral positions.

The algorithm for generating molecular conformations is similar to the manual manipulation of physical models by chemists for studying the geometry of chemical compounds. The algorithm starts with a geometrically optimized molecular structure as its initial point.

Building on this initial structure, the algorithm sequentially adjusts every bond in the molecule that can be rotated. During this procedure, the algorithm uses geometric rules designed to discard high-energy conformations, thus focusing the search on the most physically relevant configurations.

A distinguishing feature of this algorithm is its novel use of a diamond lattice framework to position the molecular structure. This fixed lattice system imposes constant angular and

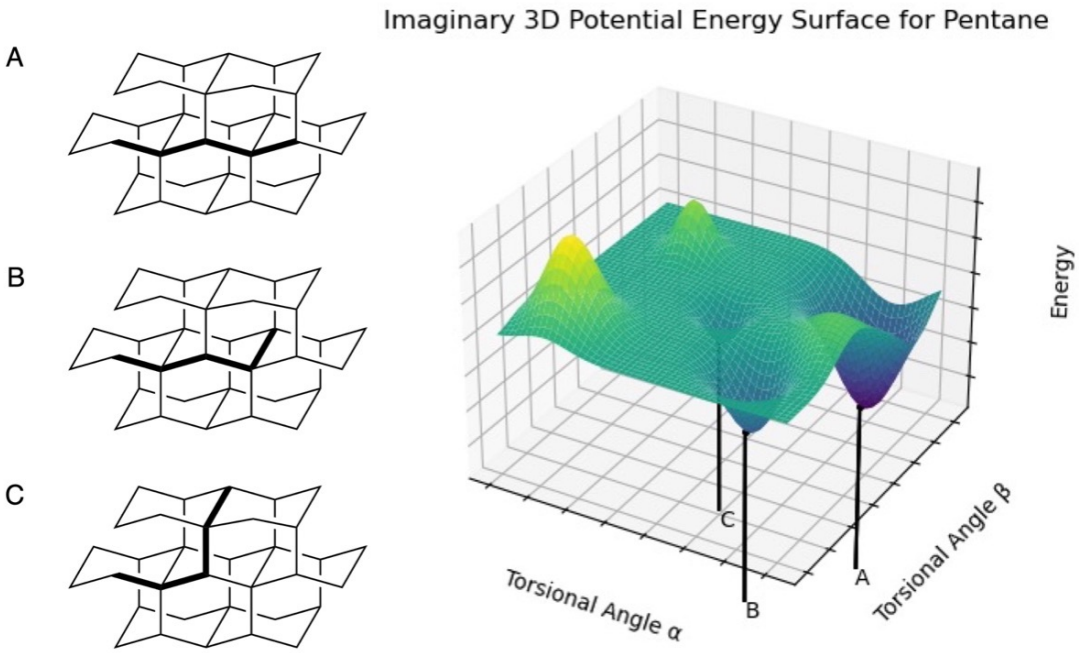


Figure 2.1 In this diamond lattice-based model, low-energy conformers can be rapidly pinpointed (Taking pentane as an example).

directional constraints on the bond rotations. Consequently, this standardization simplifies the calculations considerably.

Due to the lattice-based architecture of the algorithm, it allows for the rapid calculation of both conformational coordinates and associated geometry energy evaluations using integer arithmetic. For example, it is not necessary to perform the arithmetic manipulations required by Pythagoras' theorem to calculate distances, because all atom separations are defined by the diamond lattice. This accelerates the global search process. Furthermore, the conformations produced by the algorithm are near local energy minima, which speeds up convergence in subsequent optimization steps.

The searching strategy employs an advanced paradigm based on internal coordinate tree searches and unrestricted systematic explorations. These approaches are well-documented in the relevant literature [41, 44].

The algorithm employs a Depth-First Search (DFS) strategy, which entails a comprehensive examination of conformational options at each hierarchical level of the tree, starting from the root node and exploring as far as possible along each branch before backtracking. In the conformer torsion tree, the search begins with the torsion at the one end of the molecule, varying it through all possible values before proceeding to the next torsion closer to the 'root' or 'other end' of the molecule. In this manner, the program investigates all possible conformations, which are represented as leaf nodes in the tree structure. In this tree-based representation, the edges between nodes signify transitions between different molecular conformations, arising from variations in torsional angles. When very high energy motifs are discovered, such as two atoms on top of each other, there is no need to explore further along the molecule as all conformations with this motif must be inaccessibly high in energy. Nodes located between the root and the leaves represent molecular structures in intermediate states of torsional adjustment, awaiting further exploration.

To elucidate the search process, the molecule of pentane serves as a working example, as graphically presented in Figure 2.2. Initially, the pentane molecule is positioned within a diamond lattice framework. Within this lattice, two primary torsion angles, denoted as $\Phi 1$ and $\Phi 2$, are identified for examination. Assuming the process progresses from right to left, it would start by twisting the torsion from $\Phi 2$. The tree search operation begins at the base level (leaf nodes), which in this example corresponds to the $\Phi 2$ angle. Within the diamond lattice, each node signifies a 120° rotation of the torsion angle. Given that the molecular structure starts at its lowest energy state, marked at 180°, there is no immediate necessity to

alter this initial setting: it functions as the algorithmic origin. From this point, the angle Φ2 is systematically rotated at two increments of 120 degrees to get +60° and -60° separately, while the rest of the molecular architecture stays unaltered. Once the possibilities for Φ2 have been fully explored, the search algorithm shifts its focus to alter level 1. At this juncture, Φ1 is rotated to a new specified value and then level 2 is re-examined exhaustively under this altered configuration. This loop continues iteratively until the algorithm reaches the terminal leaf node, in this case, labeled as -Φ2=-60° in Figure 2.2. The integration of a diamond lattice framework substantially accelerates the tree-searching computational tasks, specifically in determining the coordinates crucial for generating conformers using integer calculation. This leads to enhanced computational efficiency, making the algorithm potentially robust for large-scale molecular modeling tasks.

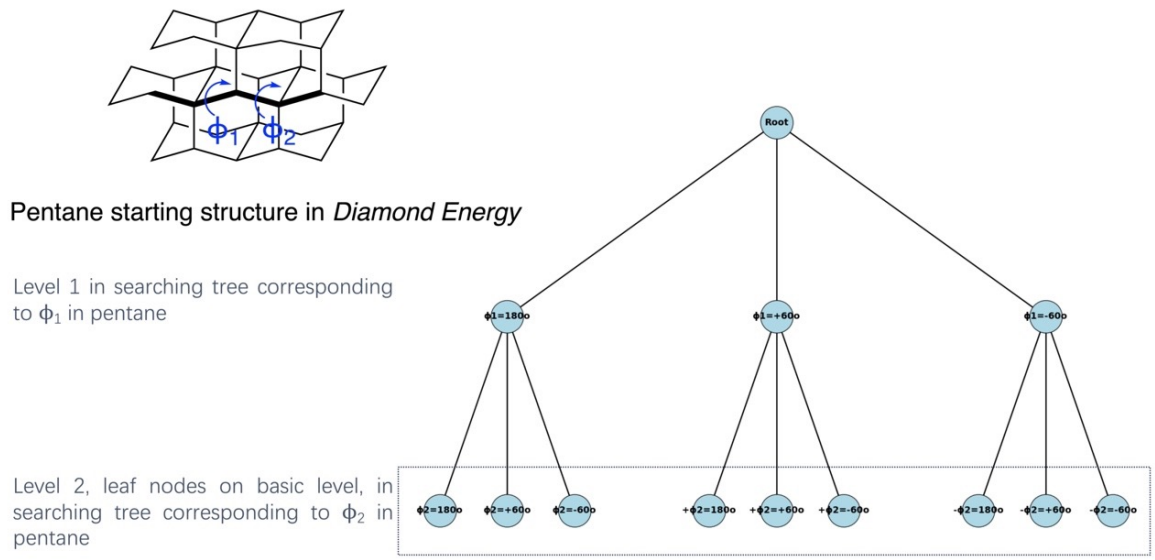


Figure 2.2 Example of a pentane searching tree in *Diamond Energy*.

For each conformation produced by the algorithm, a quick geometric assessment is conducted to evaluate its associated energy level. Conformations with energy levels exceeding a set threshold are immediately eliminated from further analysis. To expedite these evaluations, distances for non-bonded intermolecular interactions are pre-calculated within the framework of a diamond lattice. If an interaction with prohibitively high energy is detected during the tree-based search, the corresponding branch is pruned, thereby eliminating the need to explore associated sub-branches, and optimizing computational time.

The following offers a concise overview of how the algorithm evaluates the energy of generated molecular geometries. The process starts with initial conformations, which are systematically placed onto a diamond lattice under the assumption that these structures represent the lowest energy states. As the algorithm generates new conformations through the rotation of flexible bonds, it employs a set of simple, geometry-based evaluations for scoring and filtering. The primary aim of these assessments is to quickly eliminate conformations that are unlikely to evolve into low-energy, unique structural states upon further geometric optimization.

The placement of the molecular structure within a diamond lattice streamlines the computation of atomic distances, enabling rapid evaluations of all non-bonded interactions. Specifically, the program efficiently loops through the entire molecule to measure the distances between each pair of atoms. Due to the arrangement within the diamond lattice, the distances corresponding to unfavorable non-bonded interactions are predetermined. This feature allows the program to swiftly identify and discard these problematic interactions in the loop.

In the evaluation of strain energy within molecular conformations, emphasis is placed on specific types of non-bonded interactions based on non-bonded atomic distances. These interactions are categorized as 'Overlap', 'Adjacent', '1,5-Diaxial', 'Gauche', and 'Axeq (axial and equatorial positions)', each with its own energy contribution and conditions for occurrence in a diamond lattice structure. 'Overlap' interactions refer to atoms with identical coordinates and are automatically eliminated from consideration due to their prohibitive energy cost. 'Adjacent' interactions describe atom pairs separated by a single bond. These also typically have high energy contributions and are less likely to contribute to stable conformations. '1,5-Diaxial' interactions involve non-bonded atom pairs that are two bonds apart and contribute an additional 20 units (around 20 kJ/mol) to the total energy of the conformation. These are similar in structure to 1,3-diaxial interactions and should not be overlooked. 'Gauche' interactions occur when non-bonded atom pairs are separated by three bonds within a diamond lattice and have an energy contribution of approximately 2 units (around 2 kJ/mol). For atom pairs that are four bonds apart, 'Axeq' interactions become possible. These interactions contribute an additional term to the total energy of –2 units (around -2 kJ/mol). For atom pairs separated by more than five bonds, no additional energy contributions are considered, as interactions beyond this distance have already been accounted for in the energy evaluation methodology. To provide a clear picture, all of these non-bonded interactions are depicted in Figure 2.3, which elucidates their contributions to the total strain energy of molecular conformations within the diamond lattice.

Due to the atomic distances being fixed within the diamond lattice, the non-bonded interactions being assessed can be represented as integer values. This integer representation facilitates quick arithmetic calculations during the evaluation and manipulation of molecular structures, thereby improving computational efficiency. Once the process of conformational generation is complete, the resulting conformations are systematically ranked based on the total sum of their individual geometric strain energies.

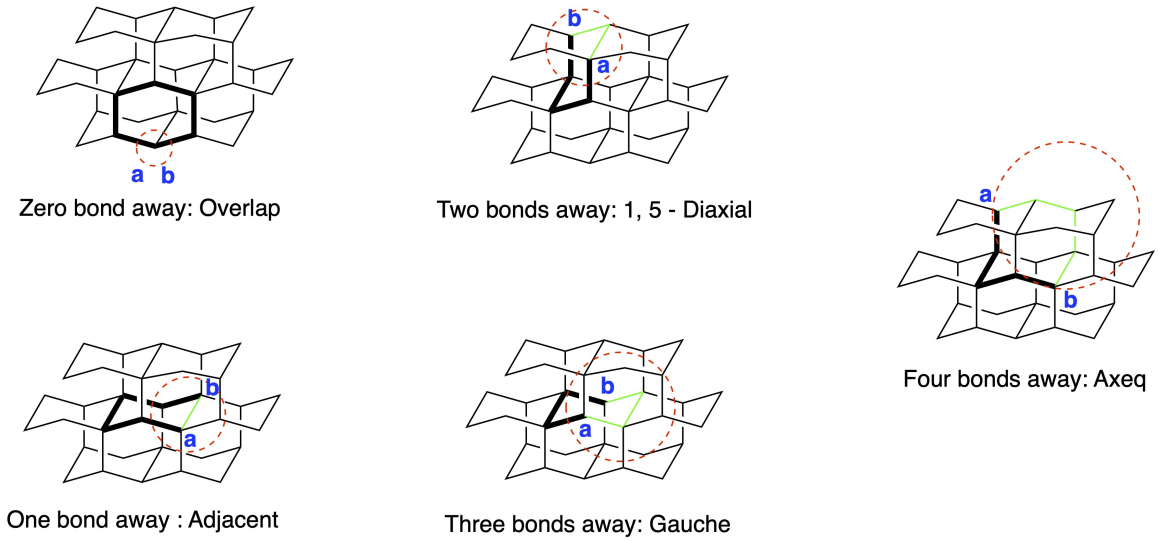


Figure 2.3 Comprehensive Representation of Non-Bonded Interactions within Diamond Lattice:This figure provides an in-depth visual of the various non-bonded interactions found in the diamond lattice's conformers. Each interaction type, from 'Overlap' to 'Axeq (axial and equatorial positions)', is categorized based on non-bonded atomic distances, and their respective energy contributions are highlighted. The terminal atoms are denoted as **a** and **b**, and the distances of their non-bonded interactions are distinctly marked in green. This illustration serves as a detailed guide, emphasizing the roles of these interactions in dictating the strain energy of molecular structures within the lattice.

2.2 Computational Details

The *Diamond Energy* program has been developed utilizing Python 3.6 or higher versions and uses subroutines from the RDKit computational environment. It is specifically designed to accept molecular representatives in the form of International Chemical Identifiers (InChI) as input. The program subsequently produces an ensemble of low-energy conformers, which are amenable to further minimization techniques using a Quantum Mechanics or Molecular

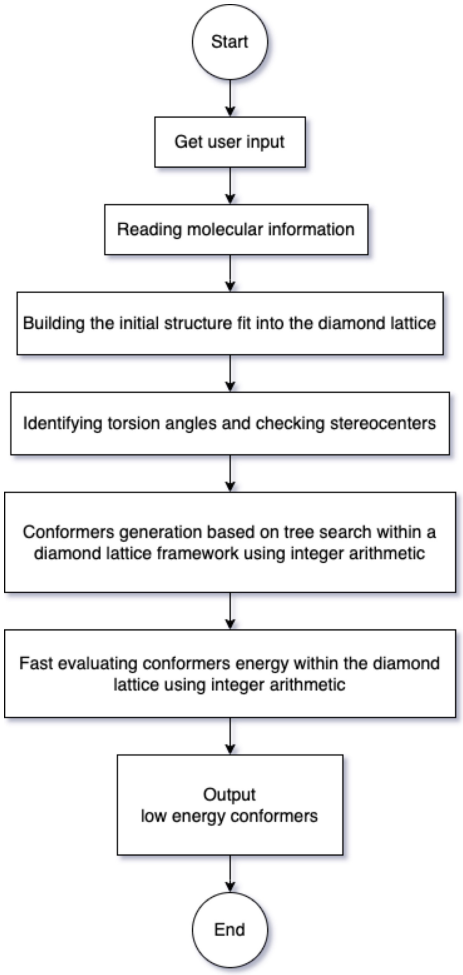


Figure 2.4 Flowchart of *Diamond Energy*.

Mechanics method. The developmental and testing phases of the software were executed on Linux and MacOS operating systems.

In the ensuing sections, Figure 2.4 will present a flowchart describing the operational architecture of the *Diamond Energy* program. This will be followed by a detailed discussion explaining the various functional aspects of the software, as guided by the flowchart.

Initially, the program accepts the InChI as input. InChI is a computer-readable, standardized identifier for chemical structures which can be used as a representation for organic systems. It is organized into distinct layers separated by forward slashes. To elucidate the complexity of the InChI format, consider the example identifier 'InChI=1S/C9H20/c1-5-7-9(4)8(3)6-2/h8-9H,5-7H2,1-4H3/t8-,9-/m0/s1.' In this specific identifier, the initial layer, labeled as 'InChI=1S,' specifies that the Standard InChI version 1 is being employed. Following this is its chemical formula C_9H_{20}. The layer commencing with '/c' provides information about the connectivity between the skeletal atoms, which are numerically described using canonical numbers determined by the InChI algorithm. Branching structures, when present, are enclosed in brackets. The subsequent '/h' layer offers details concerning hydrogen placement: carbon atoms 8 and 9 have one hydrogen attached; 5, 6 and 7 have two hydrogens; 1, 2, 3 and 4 have three hydrogens. Finally, layers indicating stereochemistry, denoted by 't8-,9-/m0/s1', record the chiral centers and molecular configuration.

Upon receiving and analyzing the molecular input data, the software begins to construct a geometric framework within a diamond lattice. The coordinates for the initial three atoms are set to a propane-like geometry, consistent with the constraints of the lattice structure. Subsequent atoms in the main chain are extended linearly, aligned with the foundational propane geometry. Atoms in any branched sub-chains are oriented along different lattice pathways. This structure construction strategy is based on empirical evidence that linear alkanes tend to adopt extended configurations [124]. At this stage, you do not need to be overly concerned with the precise arrangement of the branches. Any mistakes in chirality can be corrected later on.

One significant advantage of integrating the molecular structure into a diamond lattice framework is the improvement in computational efficiency. The algorithm employs integer arithmetic for swift calculations of spatial coordinates. To further enhance these calculations, the software employs a precalculated matrix that stores directional vectors. Each vector describes an individual atomic step within the molecular structure. Given that each atom in the lattice is capable of forming up to four covalent bonds and considering that each bond can

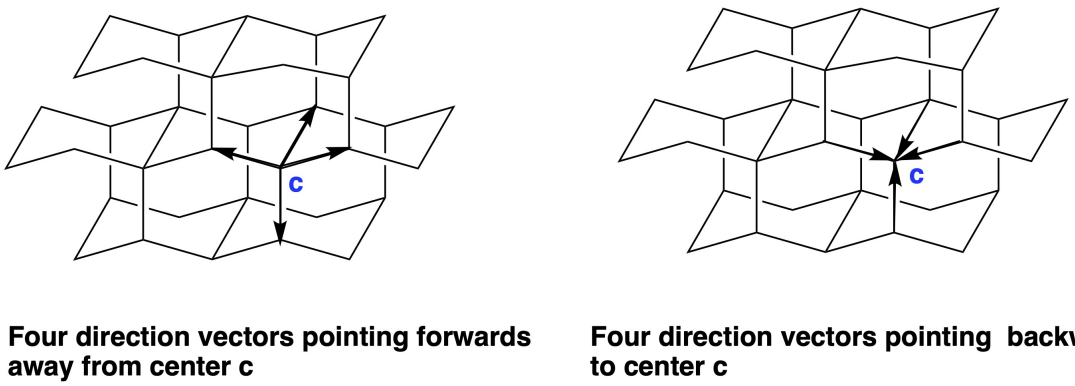


Figure 2.5 In the depicted lattice, each atom has the potential to form up to four covalent bonds. These bonds can be oriented either towards or away from a designated reference point, labeled as center **c**.

$$\begin{pmatrix} 1 & 1 & 1 \\ -1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & -1 \\ -1 & -1 & -1 \\ 1 & 1 & -1 \\ 1 & -1 & 1 \\ -1 & 1 & 1 \end{pmatrix}$$

Figure 2.6 Matrix of eight direction vectors for each step between atoms, reflecting the potential orientations of covalent bonds in the diamond lattice.

$$\begin{bmatrix} 9 & 9 & 9 & 9 & 9 & 5 & 6 & 7 \\ 9 & 9 & 9 & 9 & 4 & 9 & 6 & 7 \\ 9 & 9 & 9 & 9 & 4 & 5 & 9 & 7 \\ 9 & 9 & 9 & 9 & 4 & 5 & 6 & 9 \\ 9 & 1 & 2 & 3 & 9 & 9 & 9 & 9 \\ 0 & 9 & 2 & 3 & 9 & 9 & 9 & 9 \\ 0 & 1 & 9 & 3 & 9 & 9 & 9 & 9 \\ 0 & 1 & 2 & 9 & 9 & 9 & 9 & 9 \end{bmatrix}$$

$$\begin{bmatrix} 9 & 9 & 9 & 9 & 9 & 7 & 5 & 6 \\ 9 & 9 & 9 & 9 & 6 & 9 & 7 & 4 \\ 9 & 9 & 9 & 9 & 7 & 4 & 9 & 5 \\ 9 & 9 & 9 & 9 & 5 & 6 & 4 & 9 \\ 9 & 2 & 3 & 1 & 9 & 9 & 9 & 9 \\ 3 & 9 & 0 & 2 & 9 & 9 & 9 & 9 \\ 1 & 3 & 9 & 0 & 9 & 9 & 9 & 9 \\ 2 & 0 & 1 & 9 & 9 & 9 & 9 & 9 \end{bmatrix}$$

$$\begin{bmatrix} 9 & 9 & 9 & 9 & 9 & 6 & 7 & 5 \\ 9 & 9 & 9 & 9 & 7 & 9 & 4 & 6 \\ 9 & 9 & 9 & 9 & 5 & 7 & 9 & 4 \\ 9 & 9 & 9 & 9 & 6 & 4 & 5 & 9 \\ 9 & 3 & 1 & 2 & 9 & 9 & 9 & 9 \\ 2 & 9 & 3 & 0 & 9 & 9 & 9 & 9 \\ 3 & 0 & 9 & 1 & 9 & 9 & 9 & 9 \\ 1 & 2 & 0 & 9 & 9 & 9 & 9 & 9 \end{bmatrix}$$

Figure 2.7 A look-up table for fast calculating the true direction of the current working atom.

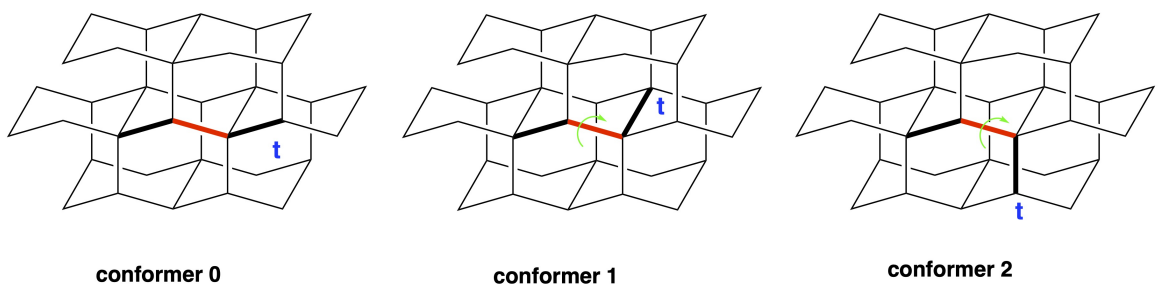


Figure 2.8 Illustrates the process employed in the *Diamond Energy* algorithm to generate and calculate the geometry of new conformers. This is done by utilizing a lookup table and a matrix of eight directional vectors.

be oriented either forwards or backwards relative to a defined reference point (as shown in Figure 2.5), the algorithm operates within a predefined set of eight unique direction vectors, as depicted in Figure 2.6.

The final position of each atom is influenced by several factors, including its attached atom in the chain, the preceding atom, and the twist of the most recent bond. Due to the diamond lattice structure's fixed patterns, we can pre-compute all possible positions and store them in a lookup table for fast access during calculations, as illustrated in Figure 2.7. It is important to note that if the system assigns a direction value of 9—surpassing the eight possible directions previously outlined for the diamond lattice framework—it indicates a potential error in the process.

Figure 2.8 using the Butane molecule as an example: I. **Conformer 0** is the initial structure created by the *Diamond Energy* algorithm. II. The first three atoms are always arranged as propane, believed to be the low-energy configuration for most alkane chains based on experience. III. In the figure, a rotatable bond is highlighted in red. IV. When generating a new conformer, the algorithm twists around this bond. For the starting structure, **Conformer 0**, the directional values assigned to the four atoms are (5, 0, 5, 0). V. To compute the next conformer, the program recalculates the terminal atom t's position. This position is influenced by its adjacent atom, the preceding atom, and the twist bond. VI. **Conformer 1**'s direction is calculated using the formula: newdirection = twist*64 + current direction of adjacent atom*8 + direction of preceding atom. Using the given values: (1*64+5*8+0), the lookup table provides a new directional value of 3 for atom **t**. It is noted that the lookup table's indexing begins at 0. VII. With this new directional value, the coordinates for atom t can be computed using integer calculations based on the previous atom's coordinates and the directional vectors. VIII. Following this process, if the bond is twisted further, **Conformer 2** is generated using the same procedure.

Once the molecular structure is fully assembled within the diamond lattice framework, the software begins an exhaustive scan to identify and record torsion angles for future rotation adjustments. This identification process systematically enumerates such angles by recognizing rotatable bonds in the molecular structure. Starting from a designated initial atom, the software sequentially moves to the fourth atom in line and identifies it as the first rotatable bond. Subsequent atoms are tested to see if they share the same previous third atom in the molecular chain. If they do, they are categorized under the existing torsion angle. If not, a new rotatable bond is added to the record. This identification and cataloging process iterates until every atom in the molecular structure has been examined and classified.

Upon finalizing the identification of torsion angles, the software transitions its focus to check the accurate configuration of stereocenters in the molecular architecture. As the molecular structure is constructed based on information provided by the InChI, the *Diamond Energy* program adheres to InChI's principles for determining chirality to ensure consistency with the input molecular data. Unlike the Cahn-Ingold-Prelog (CIP) rules that traditionally define atomic priorities for chirality assessment, InChI employs its own sequence of atomic ordering.

In the InChI algorithm, tetrahedral parity is designated as '+' when the canonical numbers of adjacent atoms increase in a clockwise manner. Unlike traditional CIP methodology, this observation is made from the perspective of a hydrogen atom or an atom with the smallest canonical number. InChI utilizes vector-based calculations to simulate human observation of molecular chirality, assessing whether atoms, when viewed from the direction of a hydrogen atom, are arranged in a clockwise or counterclockwise manner based on InChI's priority criteria.

The *Diamond Energy* script adopts a similar methodology for inspecting stereocenters within the molecular structure. For each identified stereocenter, the algorithm initially determines the attached atoms with the highest, second-highest, and lowest priorities according to InChI ordering. Directional vectors between these prioritized atoms and the corresponding stereocenter are carefully constructed. Mathematical operations, including the computation of cross and dot products, are subsequently performed using these vectors. The outcomes of these operations serve to ascertain the chirality or 'handedness' of each individual stereocenter, conforming to the standards set by the RDKit guidelines. Should any inconsistencies arise between the calculated and initial stereochemical data, the algorithm takes corrective measures by adjusting the molecular configuration to ensure accurate stereochemistry.

After all these are done, an initial structure is then ready for conformer generation. As illustrated in Figure 2.2, the algorithm employs a tree-searching methodology to generate various conformations of a molecule. The algorithm starts at searching the initial nodes in a structured tree and goes through each branch to examine all possible rotations a molecule can have. The use of a diamond lattice framework speeds up this work. This framework lets the algorithm easily adjust the molecule by looking at a prepared list. Because the framework is simple, the algorithm can use basic integer maths to quickly do the needed calculations. As the algorithm goes through the tree, the rest of the molecule which does not need to change are memorized and reused, therefore the algorithm doesn't have to repeat the same calculations, saving time and computer memory. A lookup table is also used to quickly find

out where atoms will be located after a bond is rotated. This table helps speed up the whole process.

After creating a new conformer for the molecule, the algorithm quickly checks its stability by looking at the distances between atoms within the diamond lattice framework, based on the non-bonded interactions rules mentioned before. Non-bonded atom pairs that could make the molecule unstable are identified in advance. If the algorithm detects such high-energy pairings, it promptly sums up their energy to find out the total energy of that particular arrangement. If it finds serious problems—like atoms being too close to each other—the algorithm stops exploring that branch of the tree and moves to other branches. This makes the search for stable arrangements more efficient, and low energy conformer can then be output and refined further using MM or QM.

To see how well *Diamond Energy* algorithm performs, it was compared with other top algorithms like Internal Coordinate Monte Carlo Method (MCMM) [61], Unbounded Systematic Search (SPMC) [44], Low-mode search method (LMCS) [16], and Monte Carlo combined with Low-mode search (MCLMCS) [9]. All of these algorithms are part of a software suite called MacroModel [9].

Two different types of force fields: MM3 [117] and OPLS4 [125] will be used for a fair comparison and the same optimization process for all algorithms. First, a technique called Powell-Reeves conjugate gradient [126] was applied and considered one of the best for general optimization. Then the full-matrix Newton-Raphson [9] method was used for more precise results. After that, it is confirmed that the final structure are true minima by checking their vibrational frequencies.

The optimization process was run for a maximum of 5000 steps and it stopped once it met certain criteria. Otherwise, all the parameters for other algorithms used default setting value in MacroModel.

Diamond Energy program is written in Python and works in the RDKit environment. The tests were run in batch mode using MacroModel's command line features. The comparison algorithms were also run in the same way for a fair comparison. The Internal Coordinate Monte Carlo Method used a specific mode called 'usage-directed search,' while the rest used MacroModel's default settings.

2.3 Results and Discussion

2.3.1 Evaluation Dataset for the *Diamond Energy* Algorithm

The following showcases all the molecules within the evaluation dataset, employed to assess the performance of the *Diamond Energy* algorithm (Figure 2.9).

To evaluate the performance of the *Diamond Energy* program, a diverse set of molecular structures were selected for testing. This set encompasses various types of alkanes, including isomers with distinct branching patterns and stereocenters. The aim is to rigorously examine the functionality of each component within the program. Additionally, it included linear alkenes with up to 17 rotatable bonds to assess the computational limits of the program. Molecules with exceedingly high energy, such as **17a**—an alkane unlikely to exist naturally [127]—are also incorporated in the test suite to investigate the program's predictive accuracy.

2.3.2 Comprehensive Analysis and Demonstration of *Diamond Energy*'s Operational Mechanics

Before diving into the comparative performance of the *Diamond Energy* algorithm against four other leading algorithms in a full-cycle conformer search under the MM3/OPLS4 force field, section 2.3.2 will explore several simple examples to illustrate and validate the algorithm.

Take the simple butane **c4** molecule as an initial example. Figure 2.10

The algorithm begins by reading the InChI of **c4**, given as "InChI=1S/C4H10/c1-3-4-2/h3-4H2,1-2H3". From this, the program extracts vital information such as the number of atoms excluding hydrogen (four in this case) and atom connectivity, represented by "c1-3-4-2" (InChI numbering).

Subsequently, the program constructs initial structures for **c4**. The coordinates for the first three atoms are predetermined, resembling the propane molecule (Figure 2.11). As previously noted, this approach is adopted because the molecule aligns with a diamond lattice and this alignment eliminates the need to recalculate the coordinates of the initial three atoms. Furthermore, by leveraging integer arithmetic, the program can rapidly determine the

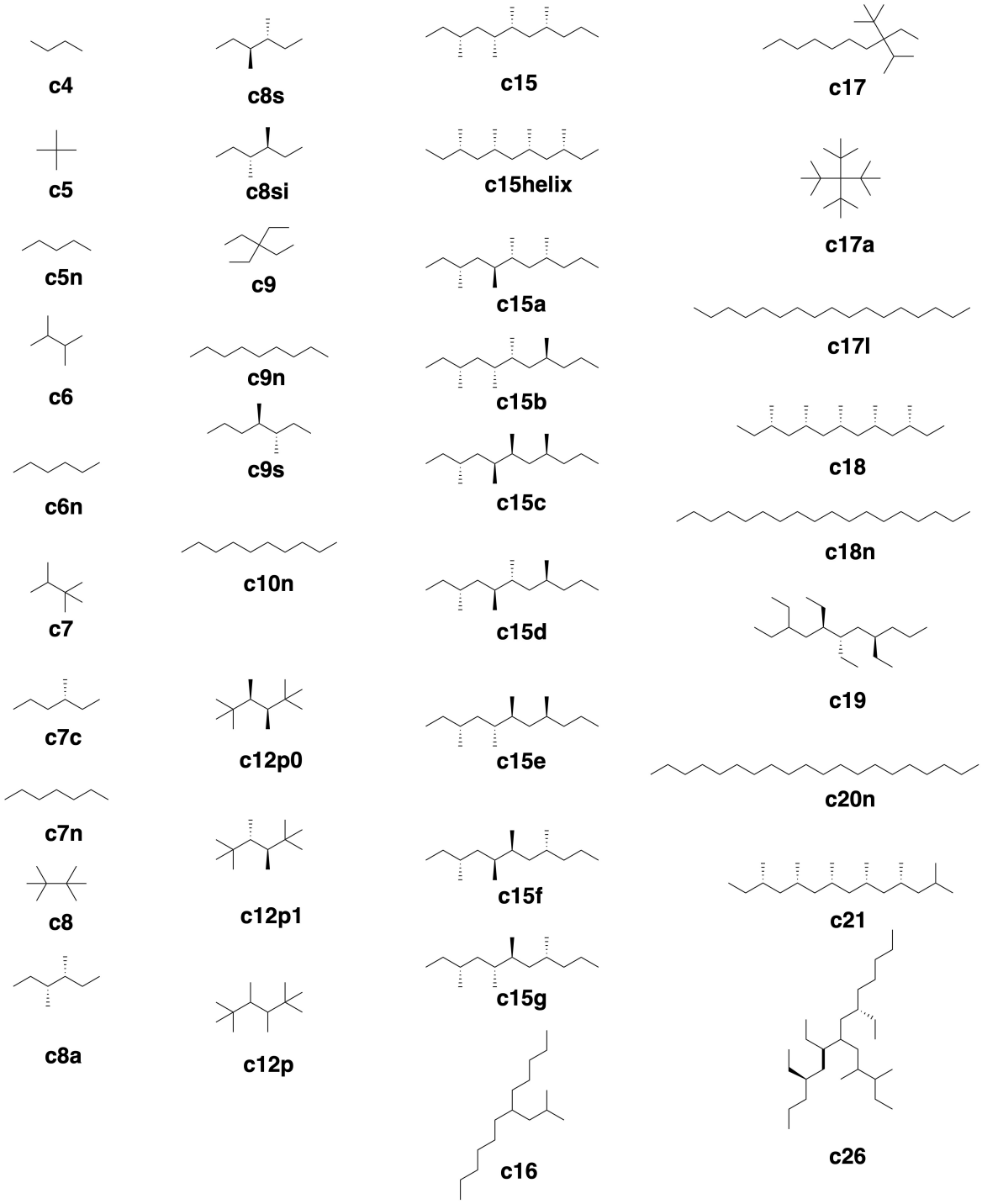


Figure 2.9 Molecules used for testing the performance of *Diamond Energy*.

c4

InChI=1S/C4H10/c1-3-4-2/h3-4H2,1-2H3

Figure 2.10 Molecule **c4** example.

coordinates for any additional atoms. By default, such extended structures are likely in the lowest energy state for linear alkanes.

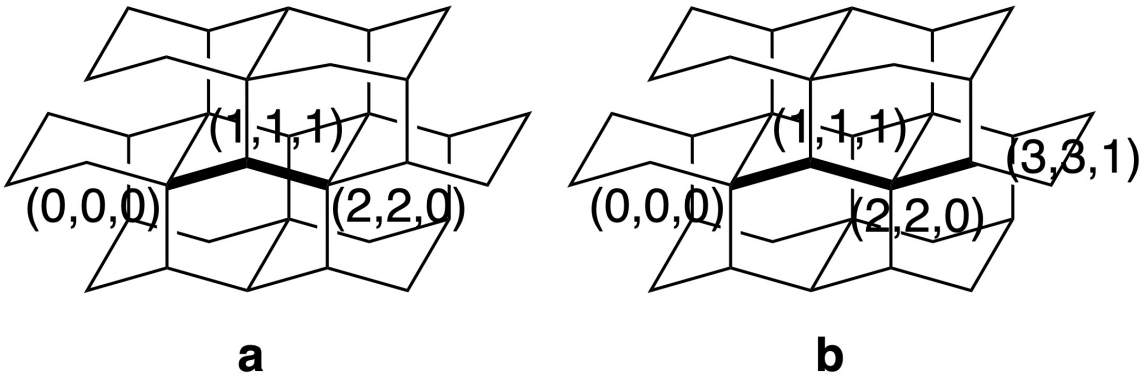


Figure 2.11 Process of Constructing Initial Structures for the **c4** Molecule: First Three Atoms Form a Fixed Propane Coordinate(a), with the Remaining Atom Extending Along the Propane Chain on a Diamond Lattice(b). The Coordinates of the Atoms Are Listed in the Brackets Beside Them.

Once the first three atoms are positioned within the diamond lattice framework, the coordinates for the fourth atom (Figure 2.11) are then determined using a lookup table, based on its attachment to the previous atom (Additionally, an explanation of how the lookup table functions can be found in Figure 2.7 and Figure 2.8). Since butane lacks a chiral center, the program bypasses stereocenter checking. After these steps, the program generates the initial structure of **c4** and logs its torsion angle information for the subsequent geometry generation stage.

The subsequent process is conformer generation. The program systematically generates new molecular structures by rotating around each rotatable bond, as cataloged previously. For the **c4** molecule, there exists a singular rotatable bond with three potential configurations, as depicted in subsequent Figure 2.12. During this structure generation, the positions of the initial three atoms remain unchanged, promoting computational efficiency in terms of calculation time and memory usage. Variations arise solely in the position of the fourth atom. Its coordinates can be determined using a pre-established lookup table. Specifically,

the position is influenced by the orientations of the linked atom and the preceding atom, in conjunction with the rotational state of the current rotatable bond.

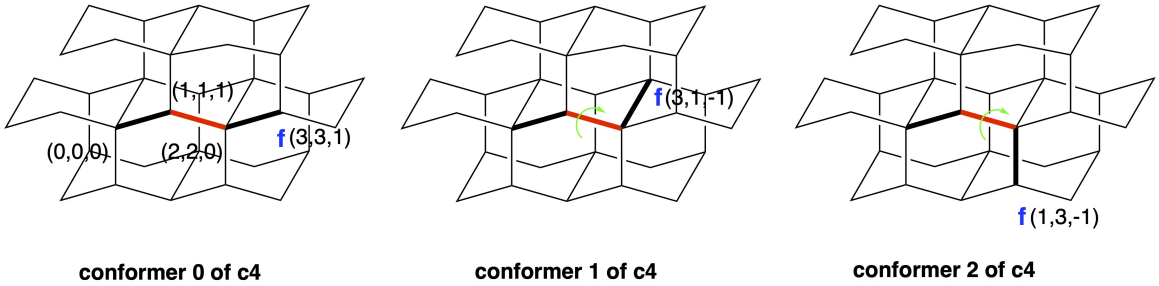


Figure 2.12 Conformer Generation in the *Diamond Energy* Algorithm for Molecule **c4**: Rotatable Bond in Red, Rotation Direction in Green, and Atom Affected by the Twist Highlighted in Blue. The Coordinates of the Atoms Are Listed in the Brackets Beside Them.

After generating each conformer, it undergoes a rapid energy evaluation. This involves iterating through the entire structure to compute atom-wise distances. As previously mentioned, specific non-bond interactions have fixed distance values within diamond lattice framework. When these patterns are detected, they contribute to the molecule's total energy. Only structures with a total energy less than 200 units, approximately 200 kJ/mol, are deemed to have low energy and are thus selected as starting points for minimization. These structures are believed to be proximate to local minima on the Potential Energy Surface (PES) and should facilitate quicker convergence to the true minima.

For molecule **c4**, the program found no high-energy interactions in its initial structure (Figure 2.13a). So, its energy was set to zero by the *Diamond Energy* algorithm. This energy is within the 200-unit range, making it a low-energy conformer. The other two conformers (Figure 2.13b and c) had one non-bond interaction: a gauche nonbond interaction, adding 2 units (about 2 kJ/mol) to their energy. Both of these conformers also had a total energy of 2 units, placing them in the low-energy category as well. To sum it up, the search for molecule **c4** gave three low-energy conformers (Figure 2.14).

Diamond Energy's output provides low-energy structures that fit into the diamond lattice for further refinement. As these structures are solely made up of carbon, they require the addition of hydrogen to be valid for further minimization. In this research, the RDKit package was used to add hydrogen to these structures. Although the RDKit documentation explains how to use the package for hydrogen addition (Ref.https://www.rdkit.org/docs/GettingStartedInPython.html#working-with-3d-molecules and Ref.https://www.rdkit.org/docs/source/rdkit.Chem.rdmolops.html), it does not detail the rules for assigning 3D coordinates to the added

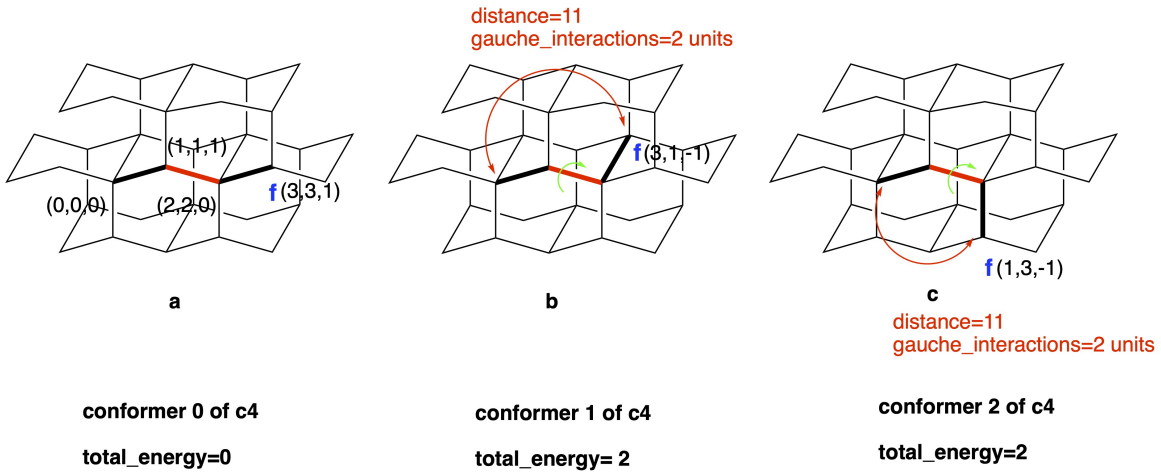


Figure 2.13 Evaluation of Conformer Energies from the *Diamond Energy* Algorithm for Molecule **c4**. (Rotatable bond: Red; Rotation direction: Green; Atoms affected by twist: Highlighted in blue; Non-bond interactions: Indicated by arrows with atom pairs, distances, and interaction types labeled in red; Atom coordinates: Provided in brackets next to them).

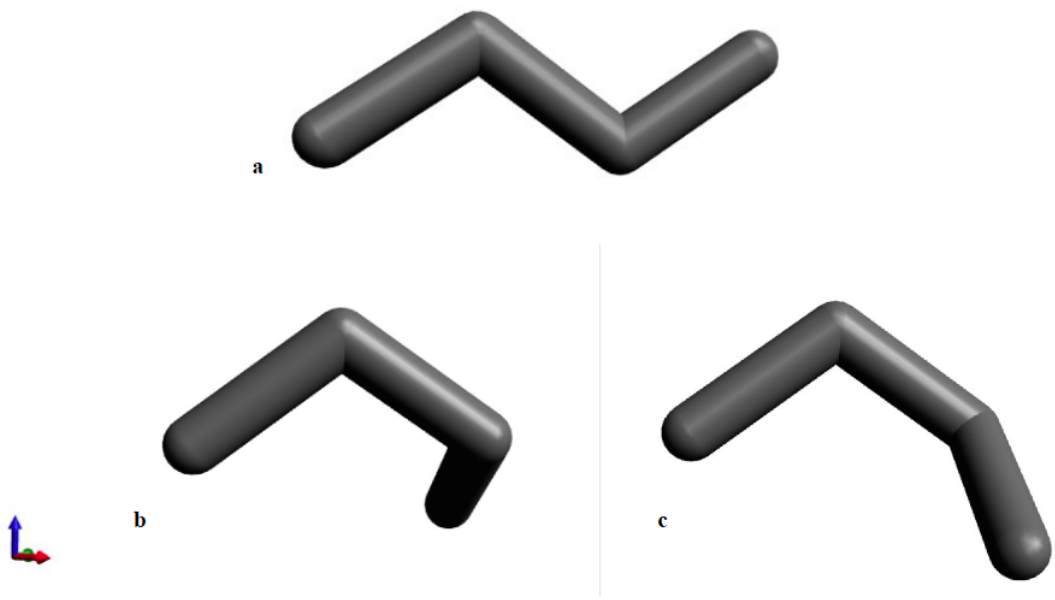


Figure 2.14 Three Low Energy Conformers in SDF Format Obtained Directly from the *Diamond Energy* Algorithm for Molecule **c4**.

hydrogens. By examining the source code (Ref.https://github.com/rdkit/rdkit/blob/master/Code/GraphMol/AddHs.cpp#L178), it is found that RDKit uses an empirical method to decide where to place a hydrogen atom, considering its position relative to nearby atoms. First, the atom's hybridization state is considered. If a hydrogen atom is being added near two other atoms, it is placed so it splits the angle between them. If it is close to three atoms that are almost in a plane, its position is determined using vector cross products. The method ensures that the atom's 3D orientation is maintained, and the hydrogen positions are based on ideal shapes without rigid calculation. The way hydrogen is added can influence the final minimized structure in subsequent experiments; detailed results on this will be presented later. This could be because adding hydrogen can direct the structure towards various close low-energy states, showing a complicated energy landscape. Additionally, since the molecules fit into a diamond lattice, the output from *Diamond Energy* might not accurately represent the actual bond lengths, as it relies on a basic scaling function. This could be why, in later force field experiments, even when *Diamond Energy* and the MCLMCS algorithm identified the same molecular framework, their minimization outcomes differed significantly.

For molecule **c4**'s low-energy conformers, hydrogen was added using the RDKit function, Chem.AddHs(mol, addCoords=True). This function adds hydrogen based on RDKit's empirical rules without altering the primary molecular framework. The resulting configurations are shown in Figure 2.15.

After hydrogen addition, the molecule was set as a starting point for force field minimization. In this research, the *Diamond Energy* algorithm's effectiveness in pinpointing low-energy conformers within the PES was assessed using MM3 and OPLS4. The outcomes were contrasted with those from Maestro MCLMCS's default algorithm, using the **c4** molecule as a reference.

Both *Diamond Energy* and MCLMCS methods yielded results that pinpointed three predominant low-energy conformers with analogous carbon chain configurations: anti (180°) and gauche (+60° and -60°), as delineated in Table 2.1. However, these conformers displayed variances in their final structures, particularly in the orientation of hydrogen atoms and their associated energy values (refer to Figure 2.16 and Table 2.1). In order to evaluate the similarity of the conformers derived from both methods, RMSD values were determined, excluding the influence of hydrogen atoms, utilizing the RDKit RMSD computation function. The findings are detailed in Table 2.1. An RMSD value less than 2 typically indicates that two structures are closely aligned, and the RMSD of two methods corresponding conformers are all fall into this range, which should indicate the two methods found similar points on

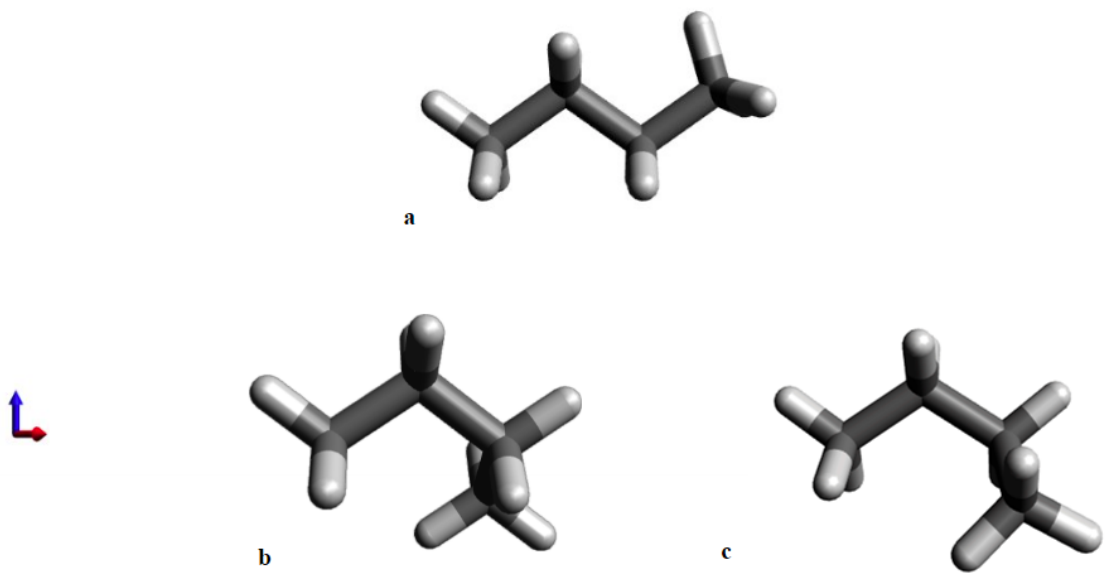


Figure 2.15 Three Low Energy Conformers Obtained from the *Diamond Energy* Algorithm for Molecule **c4** After Adding Hydrogen by RDKit in SDF Format.

PES approaching the local minima. What made them converge to different points on the PES might be due to the hydrogen handling difference and the later minimization step.

The results for **c4** using OPLS4 suggest that *Diamond Energy* effectively identifies structures near the local minima on the PES. However, ensuring that subsequent structure optimization converges to the ideal minimum is an aspect that warrants further investigation in future research.

Table 2.1 Comparison of the energy values (kcal/mol) for the three low-energy conformers of **c4** identified using the *Diamond Energy* and MCLMCS algorithms with the OPLS4 force field. The table is organized as follows: the first column lists the conformer; the second and third columns present the energy values obtained by the respective methods; and the fourth and fifth columns show their RMSD values, with and without hydrogen comparison, respectively.

Conformation	*Diamond Energy*	MCLMCS	RMSD with Hs	RMSD without Hs
Anti (180)	5.63	3.09	0.44	0.01
Gauche (60)	3.27	3.90	1.22	0.48
Gauche (-60)	3.27	3.90	1.22	0.48

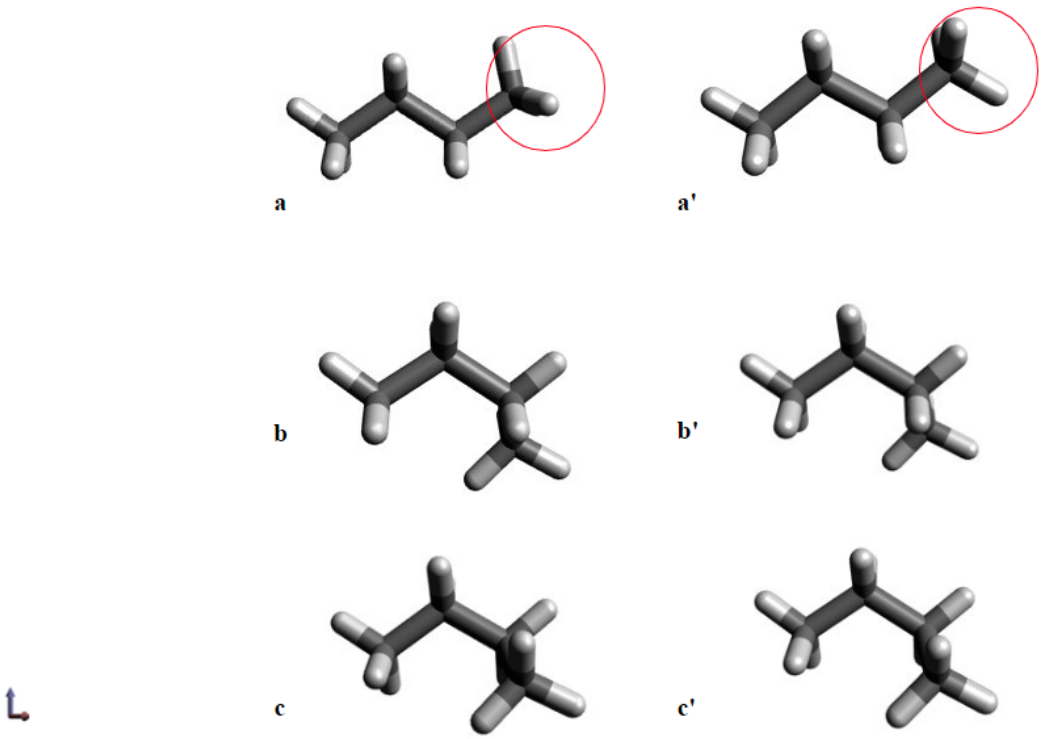


Figure 2.16 The analysis revealed three conformers of **c4** as identified by *Diamond Energy*, which were then compared to those found by MCLMCS. On the left, we have configurations from *Diamond Energy*, and on the right, the corresponding configurations from MCLMCS. The key difference between the *Diamond Energy* and MCLMCS results, particularly for the anti configuration of **c4** , lies in the orientation of the hydrogen atoms attached to the terminal carbons. The calculation were done in OPLS4 force field.

The results for MM3 of **c4** have been presented. A similar trend is observed under MM3 calculations too (Table 2.2). Based on the RMSD values, excluding hydrogen, both methods appear to identify similar geometric approaches to local minima. However, when considering the energy values from the final minimization with hydrogen, it appears that the anti-configured conformers tend toward a saddle point. The primary distinction between them is in the orientation of their hydrogens.

Table 2.2 Comparison of the energy values (kcal/mol) for the three low-energy conformers of **c4** identified using the *Diamond Energy* and MCLMCS algorithms with the MM3 force field. The table is organized as follows: the first column lists the conformer; the second and third columns present the energy values obtained by the respective methods; and the fourth and fifth columns show their RMSD values, with and without hydrogen comparison, respectively.

Conformation	*Diamond Energy*	MCLMCS	RMSD with Hs	RMSD without Hs
Anti (180)	5.78	3.09	1.58	1.21
Gauche (60)	3.90	3.90	1.28	1.21
Gauche (-60)	3.90	3.90	1.28	1.21

Both results for **c4** using OPLS4 and MM3 suggest that *Diamond Energy* effectively identifies structures near the local minima on the PES. However, ensuring that subsequent structure optimization converges to the ideal minimum is an aspect that warrants further investigation in future research. It is suggested that the PES of simple butane might be more subtle than expected and the different orientation of hydrogen could lead to falling into different wells after minimization.

An alternative explanation could be that the geometric output from *Diamond Energy* recovers bond lengths through a basic scaling function, which might not yield an exact representation of a true molecular structure. To test this hypothesis, instead of using RDKit's Chem.AddHs(mol, addCoord=True) — which only adds hydrogen in positions based on RDKit's empirical rules — another Embed function with ETKDG was utilized after adding hydrogen. This function adjusts the molecule's bond lengths and angles since ETKDG is a method that derives molecular details from a large real molecular dataset CSD. The intention was to enhance the geometric accuracy of *Diamond Energy*'s output. The results of this experiment are presented below. It seems that the Embed method excessively distorts the structure, causing the anti-configuration of the *Diamond Energy* to be overlooked in this attempt (Table 2.3 and Table 2.4).

The outcomes for **c4** indicate that *Diamond Energy* successfully identified the local minima in proximity, similar to what the prominent algorithm MCLMCS achieved. However, the

nuanced and detailed PES raises questions about the optimal starting point from this *Diamond Energy* output to converge to a local minimum. This topic warrants further exploration and research.

Table 2.3 Comparison of **c4** MM3 energy of the conformers obtained by *Diamond Energy* using ETKDG optimizing the final structures and MCLMCS Algorithm.

Conformation	*Diamond Energy*	MCLMCS
Anti (180)	3.90	3.09
Gauche (60)	3.90	3.90
Gauche (-60)	3.90	3.90

Table 2.4 Comparison of **c4** OPLS4 energy of the conformers obtained by *Diamond Energy* using ETKDG optimizing the final structures and MCLMCS Algorithm.

Conformation	*Diamond Energy*	MCLMCS
Anti (180)	3.27	3.09
Gauche (60)	3.27	3.90
Gauche (-60)	3.27	3.90

Table 2.5 Comparison of the energy values (kcal/mol) for the three low-energy conformers of **c5** identified using the *Diamond Energy* and MCLMCS algorithms with the MM3 force field. The table is organized as follows: the first column lists the conformer; the second and third columns present the energy values obtained by the respective methods; and the fourth and fifth columns show their RMSD values, with and without hydrogen comparison, respectively.

Energy	*Diamond Energy*	MCLMCS	RMSD with Hs	RMSD without Hs
MM3	11.53	4.30	2.18	0.96
OPLS4	−4.41	4.30	2.18	0.96

c5

InChI=1S/C5H12/c1-5(2,3)4/h1-4H3

Figure 2.17 Molecule **c5** example.

Another sample that could assist in evaluating *Diamond Energy*'s performance is **c5**, which has a highly branched structure in Figure 2.17.

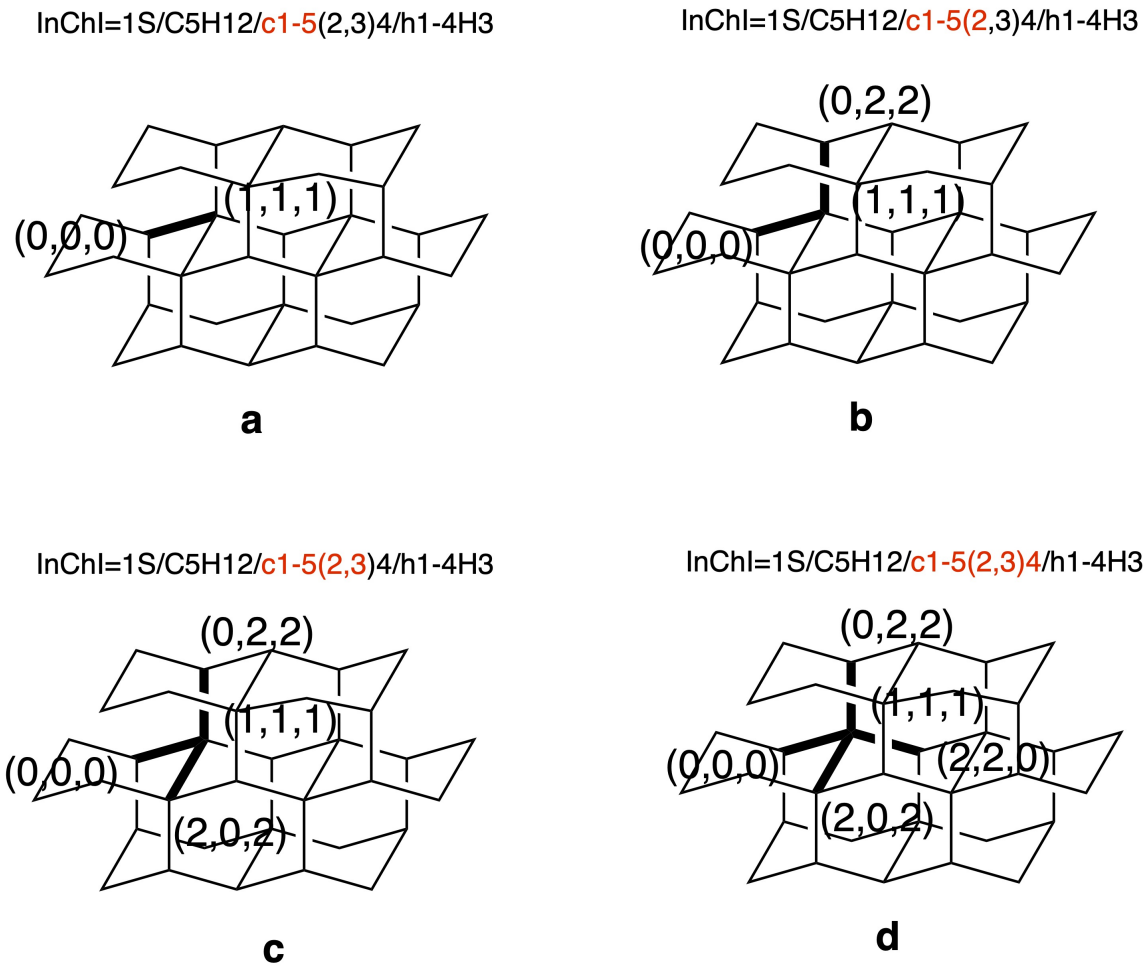
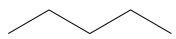


Figure 2.18 Sequence of Steps for *Diamond Energy* to Construct the **c5** Structure within a Diamond Lattice, Following InChI Connectivity Information. InChI Atoms involved in each step are highlighted in red.



c5n

InChI=1S/C5H12/c1-3-5-4-2/h3-5H2,1-2H3

Figure 2.19 Molecule **c5n** example.

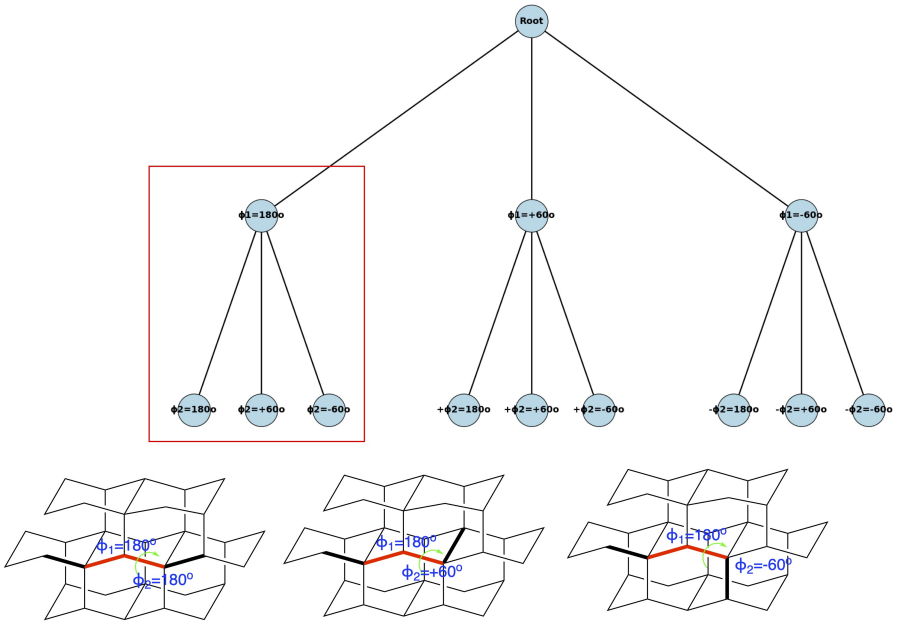


Figure 2.20 Illustrating the method employed by *Diamond Energy* for conducting a conformer search on the **c5n** molecule (pentane) by utilizing a tree search algorithm.

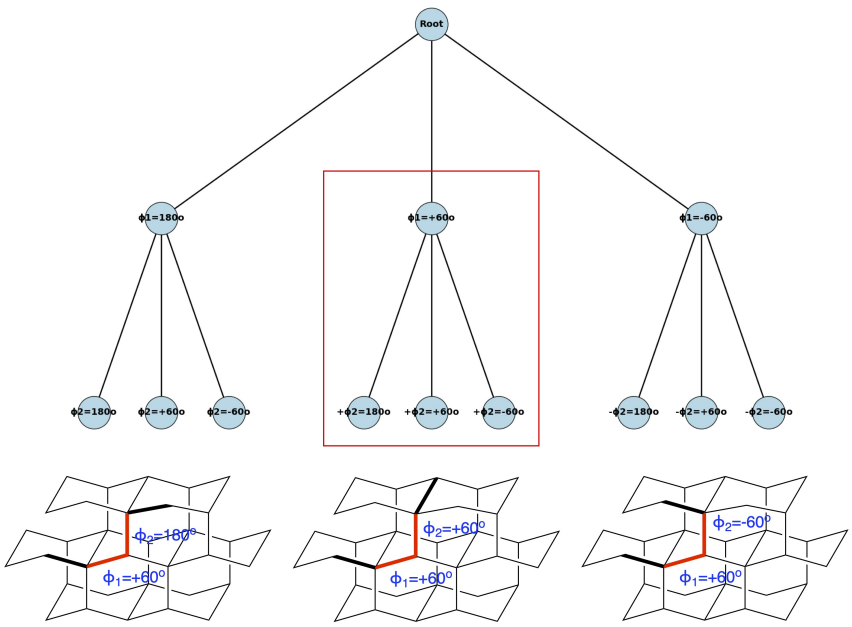


Figure 2.21 Illustrating the method employed by *Diamond Energy* for conducting a conformer search on the **c5n** molecule (pentane) by utilizing a tree search algorithm.

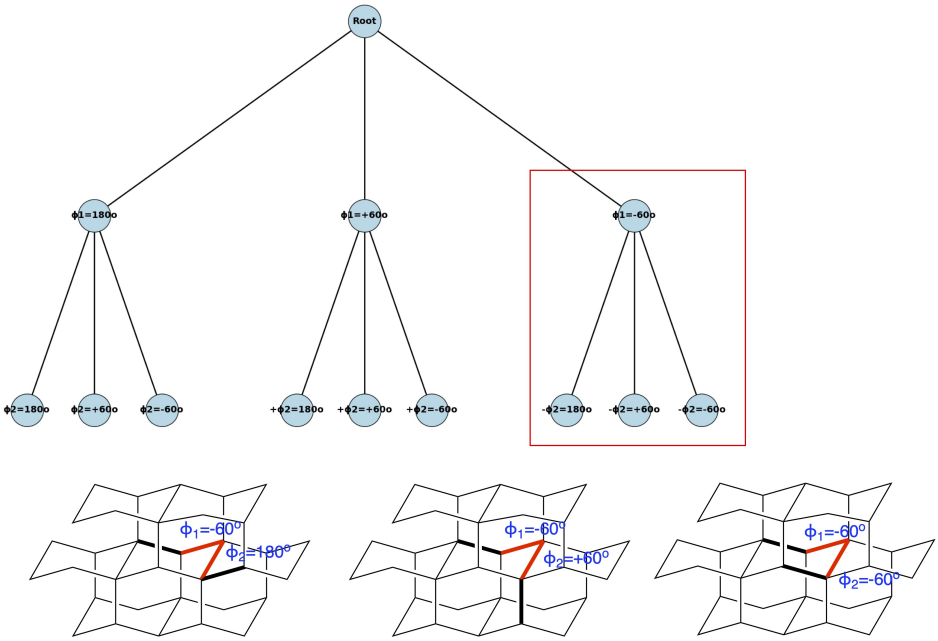


Figure 2.22 Illustrating the method employed by *Diamond Energy* for conducting a conformer search on the **c5n** molecule (pentane) by utilizing a tree search algorithm.

As in the case of **c4**, the program begins by accepting its InChI as input, processing and storing the molecular data from the InChI. Following this, the program constructs structures based on the connectivity information "c1-5(2,3)4" found in the **c5** InChI(InChI=1S/C5H12/c1-5(2,3)4/h1-4H3). Similar to the previous steps, the program establishes the first three atoms on fixed coordinates — in this scenario, atoms 1, 5, and 4 (using InChI numbering). With two branching atoms, atom 2 and atom 3, the program positions them in a manner compatible with the diamond lattice framework. Given the absence of rotatable bonds in this molecule, the program skips the conformer generation phase and focuses solely on energy evaluation. The energy assessment mirrors the procedure outlined for **c5**: a straightforward atom-wise approach to quickly identify non-bond interactions that might elevate the energy, summing up the individual energy components to derive the molecule's overall energy. Specifics will not be dived into here, but one can refer to the section 2.3.2 discussing the **c4** molecule for details. Additionally, as this molecule lacks a stereocenter, the relevant code segment is not employed as well (Figure 2.18).

In both MM3 and OPLS4 calculations, each method identified only one conformer. Similar to the **c4** case, the carbon frameworks produced by both methods were alike, yet the outcomes post-minimization, with hydrogen addition, varied. The results are summarized in Table 2.5.

A key aspect of the *Diamond Energy* program is its efficient memory usage. Instead of allocating new memory for generated structures, the program repurposes existing information when parts of the structure remain unchanged, thereby conserving memory. To demonstrate this feature, molecule **c5n** will be used as an example (Figure 2.19).

Like the earlier case, the software starts by importing the **c5n** molecule's data from the provided InChI. It then generates the initial molecular structures in a straight chain. The next step involves pinpointing the bonds in the molecule that can rotate and documenting them. After this, the software proceeds with a depth-first search through a tree-like structure (Figure 2.20, Figure 2.21 and Figure 2.22).

The program does a systematic depth-first search, structured like a tree. This search initiates at the molecule's rightmost end and progresses to the left, rotating each bond in turn. During this phase, the coordinates for the left side structures that have not been rotated are preserved to avoid redundant recalculations. For the atoms linked to any rotated bond, the software quickly calculates their new positions using directional vectors and a lookup table.

As demonstrated in Figure 2.20, Figure 2.21 and Figure 2.22 for the **c5n** (pentane) molecule, the rotation angle $\Phi 1$ remains fixed initially, while the positions of atoms associated with $\Phi 2$ are recalculated. Following the creation of all possible conformers through this method, which yields a total of nine different conformers, the program proceeds to assess their energies. This energy evaluation step follows the same process as described previously, which is not detailed again here.

What we discuss here are the comparative results for the conformer search conducted by *Diamond Energy* and MCLMCS using the MM3 force field. In this particular case, *Diamond Energy* successfully identified many energy minima, surpassing the results obtained by MCLMCS. *Diamond Energy* was able to identify more conformations, which MCLMCS did not find, demonstrating a more comprehensive search outcome. Unlike previous examples, the issue with hydrogen was not encountered in this **c5n** scenario.

Table 2.6 lists the MM3 energy values in kcal/mol for each conformation of **c5n** found by *Diamond Energy* and MCLMCS, while Table 2.7 below lists the OPLS4 energy values in kcal/mol for each conformation of **c5n** found by *Diamond Energy* and MCLMCS.

The outcomes from OPLS4 indicate that *Diamond Energy* has outperformed MCLMCS in searching for conformers of **c5n**. This is evidenced by *Diamond Energy* finding a conformer

Table 2.6 Results of **c5n** found by *Diamond Energy* and MCLMCS in MM3 (kcal/mol).

Conformation	*Diamond Energy*	MCLMCS
0 (linear)	4.14	4.14
1	5.00	5.00
2	5.00	5.00
3	5.00	5.76
4	5.00	5.76
5	5.76	7.46
6	5.76	7.46
7	7.64	
8	7.64	

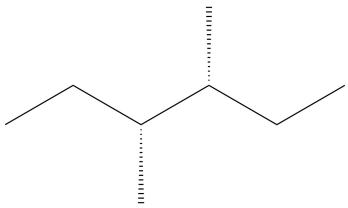
Table 2.7 Results of **c5n** found by *Diamond Energy* and MCLMCS in OPLS4 (kcal/mol).

Conformation	*Diamond Energy*	MCLMCS
0 (linear)	3.44	4.14
1	3.95	5.00
2	3.95	5.00
3	3.95	5.76
4	3.95	5.76
5	4.36	7.46
6	4.36	7.46
7	6.31	
8	6.31	

with significantly lower energy and identifying more local minima compared to the results using MM3 and OPLS4.

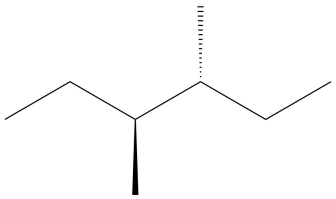
The examples provided earlier did not include checks for stereo centers. To shed light on how this segment of the code functions, two more examples are presented to demonstrate the process of detecting and rectifying inaccurately constructed chiral centers. Take the cases of **c8a** and **c8s** (Figure 2.23). The initial structure-building approach, recording of rotatable bonds, and subsequent energy evaluation remain consistent with previous examples. However, since these molecules possess stereo centers, the program, upon generating the initial structure, will conduct a check for chiral centers and correct any identified discrepancies.

The additional functionalities of the program, which include the construction of initial structures, the generation of geometries, and the evaluation of energies, have been thoroughly illustrated in prior examples. Consequently, these processes are not extensively depicted for the molecules **c8a** and **c8s**. What follows is a description of how *Diamond Energy*'s stereocenter checking mechanism operates.



c8a

InChI=1S/C8H18/c1-5-7(3)8(4)6-2/h7-8H,5-6H2,1-4H3/t7-,8-/m1/s1



c8s

InChI=1S/C8H18/c1-5-7(3)8(4)6-2/h7-8H,5-6H2,1-4H3/t7-,8+

Figure 2.23 Molecule **c8a** and **c8s** examples.

Once the initial structure is assembled, the program examines each stereocenter in the order specified by the InChI, mirroring the sequence used to create the initial atomic structure. During the initial build phase, the program does not concern itself with the stereochemistry. It is only in the subsequent checking phase that it addresses any inaccuracies. If an incorrectly constructed chiral center is detected, the program will adjust the structure to rectify this center. This correction may necessitate recalculating the affected structures while leaving the previously correct atoms unchanged. The program will continue this checking process until all recorded stereocenters have been verified, ensuring that all chiral centers are accurately represented.

The molecules **c8a** and **c8s** are utilized to demonstrate the process of checking and adjusting the configuration of stereocenters in *Diamond Energy*. These molecules are identical except

for the arrangement of two stereocenters. By default, the program obtains two molecules with identical structures, as depicted in Figure 2.24, the molecule **c8s** correctly displays both stereocenters 7 and 8, referred to as S7 and S8 following InChI numbering and marked accordingly in the figure. In contrast, molecule **c8a** has an incorrectly represented stereocenter at position S7. The process of stereochecking involves identifying the three largest atoms by their InChI numbers rather than using CIP rules. The program then calculates the dot and cross products using these atoms and their central atom. These calculations help determine the spatial orientation of the atoms, whether it's in a clockwise or counterclockwise configuration. Consequently, **c8s** moves forward to the conformer search phase like earlier examples, whereas **c8a** must have its S7 stereocenter error corrected. This involves reconstructing its structure, along with any other atoms impacted by this change, before it can move forward.

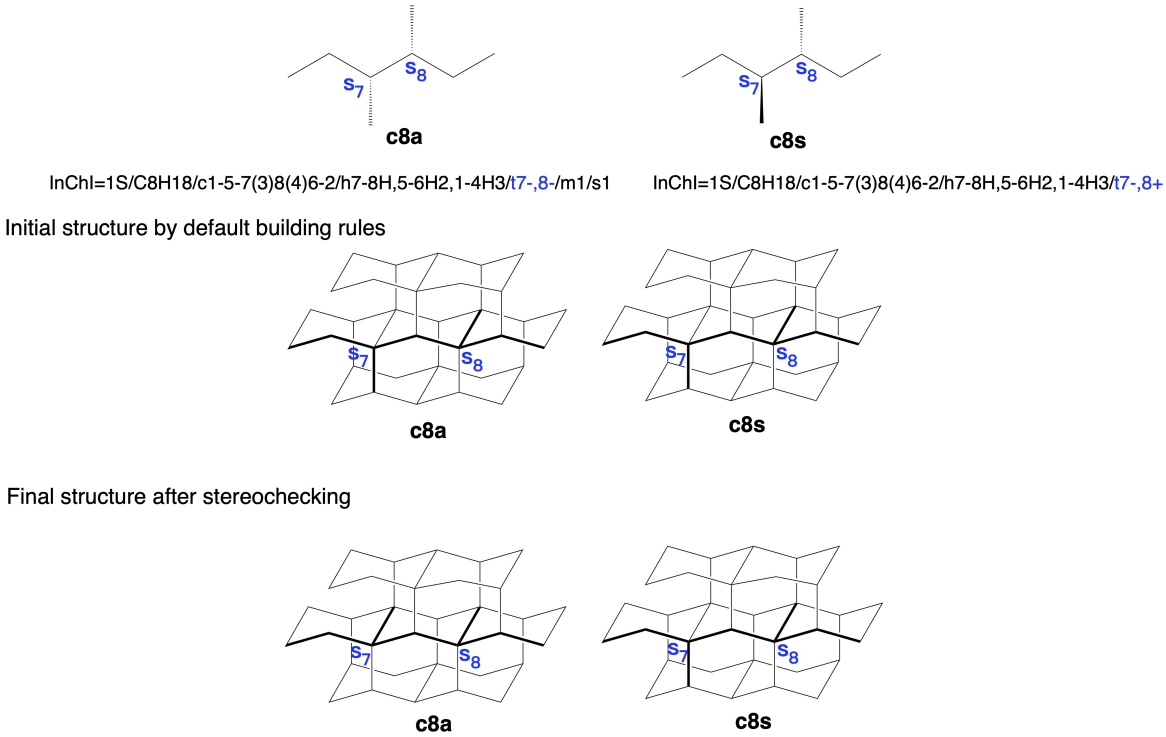


Figure 2.24 This figure illustrates the molecules **c8a** and **c8s**, which are identical except for two opposite stereocenters. The program's default configuration initially assigns the same structure to both molecules. This leads to an accurate structure for **c8s**, but an error in the stereocenter for **c8a**. Consequently, **c8s** moves forward to the conformer search phase, as shown in previous examples, whereas **c8a** requires adjustments to correct the stereocenter discrepancy.

In the test cases involving **c8a** and **c8s**, it has been observed that the *Diamond Energy* algorithm identifies most local minima similarly to MCLMCS. Both approaches explore comparable low-energy regions when utilizing the MM3 and OPLS4 forces field.

Table 2.8 lists the MM3 energy values in kcal/mol for each conformation of **c8a** found by *Diamond Energy* and MCLMCS. Table 2.9 lists the OPLS4 energy values in kcal/mol for each conformation of **c8a** found by *Diamond Energy* and MCLMCS.

Table 2.8 The MM3 results of **c8a** by *Diamond Energy* and MCLMCS (kcal/mol).

Conformation	*Diamond Energy*	MCLMCS
0	13.02	13.02
1	13.02	13.02
2	13.06	13.06
3	13.06	13.06
4	13.69	13.69
5	13.69	13.69
6	13.85	13.85
7	13.88	13.88
8	15.01	15.01
9	15.01	15.01
10	15.21	15.21
11	15.21	15.21
12	15.51	15.51
13	15.51	15.51
14	15.51	15.51
15	15.51	15.51
16	15.73	15.73
17	*	*

Table 2.9 lists the OPLS4 energy values in kcal/mol for each conformation of **c8a** found by *Diamond Energy* and MCLMCS.

Table 2.9 The OPLS4 results of **c8a** by *Diamond Energy* and MCLMCS (kcal/mol).

Conformation	*Diamond Energy*	MCLMCS
0	3.97	3.97
1	3.97	3.97
2	4.03	4.03
3	4.03	4.03
4	4.71	4.71
5	*	*

Table 2.10 list and Table 2.11 the MM3 and OPLS4 energy values in kcal/mol for each conformation of **c8s** found by *Diamond Energy* and MCLMCS.

Table 2.10 The MM3 results of **c8s** by *Diamond Energy* and MCLMCS (kcal/mol).

Conformation	*Diamond Energy*	MCLMCS
0	13.00	13.00
1	13.03	13.03
2	13.06	13.06
3	13.06	13.07
4	13.07	13.67
5	13.67	13.82
6	*	*

Table 2.11 The OPLS4 results of **c8s** by *Diamond Energy* and MCLMCS (kcal/mol).

Conformation	*Diamond Energy*	MCLMCS
0	3.80	3.80
1	3.81	3.81
2	3.83	3.83
3	3.86	3.86
4	3.86	4.48
5	4.48	4.72
6	4.72	5.40
7	*	*

The study on molecules **c4**, **c5**, **c5n**, **c8a** and **c8s** indicates that the *Diamond Energy* algorithm is capable of identifying low-energy conformers for carbon chains, mirroring the efficacy of the MCLMCS algorithm. This is supported by RMSD comparisons of carbon chain conformers, both with and without hydrogen atoms. Findings suggest that the energy landscapes of the test molecules are considerably more complex and detailed than initially assumed.

To enhance the precision of the results, the study incorporates the hydrogen addition method used by MacroModel. The comparison will be made between the hydrogen addition results from RDKit and those from MacroModel. Details on hydrogen handling are not explicitly covered in the foundational papers of MCLMCS, nor in the provided MacroModel documentation, leaving a gap in detailed procedural understanding.

According to the MacroModel 10.8 Reference Manual (https://gohom.win/ManualHom/Schrodinger/Schrodinger_2015-2_docs/macromodel/macromodel_reference_manual.pdf) three methods for managing hydrogen are outlined. The first is manually adding hydrogen to molecules through the Maestro interface; however, this is unsuitable for the automated pipeline process, which is going to be used in our study later. The second method involves using the HADD keyword in a command-line script to run the MacroModel process, but the use of the HADD keyword within .com files has been tried, and it did not yield the expected results. The final process is to call the "applyhtreat" utility using a Python script, which worked in testing and should be suitable for building an automated pipeline for our study. Therefore, we choose the third method.

The alternative method converts the output from *Diamond Energy* to the .mae format, which is compatible with MacroModel jobs. The "applyhtreat" utility from MacroModel is then utilized to add hydrogen to the .mae file (https://gohom.win/ManualHom/Schrodinger/Schrodinger_2015-2_docs/general/general_utilities.pdf). This approach has been included for assessment, aiming to determine the influence of different hydrogen addition methods on the final results. The attempt aims to minimize the discrepancies between the energy landscapes produced by *Diamond Energy* and other algorithms, focusing on their treatment of hydrogen atoms. However, without detailed insights into the respective hydrogen handling algorithms, it becomes challenging to ensure that the comparison is conducted with thorough rigor. The subsequent paragraphs detail these outcomes.

Table 2.12 shows that the MacroModel way of handling hydrogen can produce the same results for MCLMCS and *Diamond Energy*, after a conformer search and energy minimization

cycle in MM3 force field. However, the RDKit way of handling hydrogen and the anti configuration seem not the same as what MCLMCS found, which can be seen in Table 2.13, first row third column. Where the RMSD for the results of *Diamond Energy* without hydrogen consideration is close to zero, which means they are almost the same in the carbon chain. However, after adding hydrogen, the value increases a bit, indicating that the structure is quite different. The first row is for the anti configuration of **c4**, and the RMSD of the MacroModel handle hydrogen, with and without hydrogen, does not have an apparent gap.

Table 2.12 The MM3 energy values in kcal/mol for each conformation of **c4** found by *Diamond Energy* using RDKit hydrogen postprocessing pipeline, *Diamond Energy* using MacroModel hydrogen postprocessing pipeline and MCLMCS (kcal/mol).

Conformation	Diamond Energy–RDKit	Diamond Energy–MacroModel	MCLMCS
Anti (180)	5.78	3.09	3.09
Gauche (60)	3.90	3.90	3.90
Gauche (-60)	3.90	3.90	3.90

Table 2.13 RMSD Comparison for Each Conformation of **c4** Identified by *Diamond Energy* Using Different Hydrogen Addition Strategies and Compared with MCLMCS in MM3 calculations. This table compares the three low-energy conformers of **c4** as identified by *Diamond Energy* through two pipelines, alongside their RMSD comparisons with MCLMCS results. The table is organized as follows: the first column lists the conformer; the second and third columns display the RMSD values of the RDKit post-processing results in comparison with MCLMCS results, with and without the addition of hydrogen, respectively; the fourth and fifth columns present the RMSD values of the MacroModel post-processing results, with and without hydrogen comparison with MCLMCS, respectively.

Conformation	Hs–RDKit	Without Hs–RDKit	Hs–MacroModel	Without Hs–MacroModel
Anti (180)	0.44	0.01	0.00	0.00
Gauche (60)	1.23	0.48	1.23	0.48
Gauche (-60)	1.23	0.48	1.23	0.48

In the case of compound **c5**, an improvement in the final results is observed when the MacroModel method is applied to adjust the output structures obtained from *Diamond Energy* calculations, before they are subjected to minimization using MM3. This is evidenced by Tables 2.14 and Table 2.15, where the final energy values are consistent between the MacroModel and MCLMCS outputs, unlike the final results produced by RDKit. Moreover, Table 2.15 indicates that structures consisting solely of carbon atoms are nearly identical when comparing outputs from *Diamond Energy* and MCLMCS. However, when hydrogen atoms are incorporated, the structures modified via the MacroModel approach maintain a high

degree of similarity to the MCLMCS results, in contrast to the RDKit-modified structures which show a greater deviation from the original carbon chain upon hydrogen addition.

Table 2.14 The MM3 energy values in kcal/mol for each conformation of **c5** found by *Diamond Energy* RDKit post-processing pipeline, *Diamond Energy* MacroModel post-processing pipeline and MCLMCS (kcal/mol).

Conformation	Diamond–RDKit	Diamond–MacroModel	MCLMCS
0	11.53	4.30	4.30

Table 2.15 RMSD Comparison for Each Conformation of **c5** Identified by *Diamond Energy* Using Different Hydrogen Addition Strategies and Compared with MCLMCS in MM3 calculations. This table compares the three low-energy conformers of **c5** as identified by *Diamond Energy* through two pipelines, alongside their RMSD comparisons with MCLMCS results. The table is organized as follows: the first column lists the conformer; the second and third columns display the RMSD values of the RDKit post-processing results in comparison with MCLMCS results, with and without the addition of hydrogen, respectively; the fourth and fifth columns present the RMSD values of the MacroModel post-processing results, with and without hydrogen comparison with MCLMCS, respectively.

Conformation	Hs–RDKit	Without Hs–RDKit	Hs–MacroModel	Without Hs–MacroModel
0	0.59	0.02	0.00	0.00

In contrast, for other compounds such as **c5n**, **c8a**, and **c8s**, the methods of hydrogen handling by RDKit or MacroModel prior to the MM3 minimization process appear to have minimal impact on the final results. For these molecules, the end results obtained from *Diamond Energy* and MCLMCS are nearly identical, indicating that both approaches are effective at identifying the lowest energy conformers within the MM3 energy landscape. This is further illustrated in the subsequent tables (Table 2.16, Table 2.17 and Table 2.18).

In every test case examined, *Diamond Energy* demonstrated a capability to navigate the low energy conformational landscape that is comparable to that of the established MCLMCS algorithm. The preprocessing treatment of structures, particularly the addition of hydrogen atoms before minimization, has been shown to influence the final outcomes in certain instances. This suggests that the energy landscapes of the molecules under study are more nuanced and complex than expected. Detailed insights into *Diamond Energy*'s effectiveness as a conformational search tool, its operational efficiency, and its precision in mapping the low energy terrains of molecules on a broader scale are discussed in the subsequent sections.

Table 2.16 The MM3 energy values in kcal/mol for each conformation of **c5n** found by *Diamond Energy* RDKit post-processing pipeline, *Diamond Energy* MacroModel post-processing pipeline and MCLMCS (kcal/mol).

Conformation	Diamond–RDKit	Diamond–MacroModel	MCLMCS
0	4.14	4.14	4.14
1	5.00	5.00	5.00
2	5.00	5.00	5.00
3	5.00	5.00	5.76
4	5.00	5.00	5.76
5	5.76	5.76	7.46
6	5.76	5.76	7.46
7	7.64	7.64	7.46
8	*	*	*

Table 2.17 The MM3 energy values in kcal/mol for each conformation of **c8a** found by *Diamond Energy* RDKit post-processing pipeline, *Diamond Energy* MacroModel post-processing pipeline and MCLMCS (kcal/mol).

Conformation	Diamond–RDKit	Diamond–MacroModel	MCLMCS
0	13.02	13.02	13.02
1	13.02	13.02	13.02
2	13.06	13.06	13.06
3	13.06	13.06	13.06
4	13.69	13.69	13.69
5	13.69	13.69	13.69
6	13.85	13.85	13.85
7	13.88	13.88	13.88
8	*	*	*

Table 2.18 The MM3 energy values in kcal/mol for each conformation of **c8s** found by *Diamond Energy* RDKit post-processing pipeline, *Diamond Energy* MacroModel post-processing pipeline and MCLMCS (kcal/mol).

Conformation	Diamond–RDKit	Diamond–MacroModel	MCLMCS
0	13.00	13.00	13.00
1	13.03	13.03	13.03
2	13.06	13.06	13.06
3	13.06	13.06	13.07
4	13.07	13.07	13.67
5	13.67	13.67	13.82
6	13.67	13.67	13.86
7	13.82	13.82	14.94
8	*	*	*

2.3.3 Assessing *Diamond Energy*'s Overall Speed and Capability for Geometry Creation

An initial execution of the *Diamond Energy* algorithm functions as a quantitative assessment of its capability to generate molecular conformations. The ensuing bar graph (Figure 2.25), illuminates two crucial metrics: the blue bars represent the aggregate number of structures that *Diamond Energy* processed during its computational search, while the red bars delineate the quantity of low-energy conformations (low energy is defined as having an energy level within a range of 200 units in *Diamond Energy* algorithm approaching 200 kJ/mol) that were efficiently assessed and deemed viable within the constraints of the diamond lattice framework. The test molecules are enumerated along the x-axis and are visually represented in Figure 2.9. It is significant to note that molecules possessing more than ten rotatable bonds have the potential to yield up to 0.1 billion conformers. To address the visual challenge this presents, the y-axis has been scaled for ease of interpretation, and exact numerical values are displayed above each respective bar for enhanced clarity.

Two particular cases demand extra attention:

The lack of a red bar for the molecule **c17a** indicates that *Diamond Energy* found this structure too high-energy to be feasible, corroborating existing scientific literature that similarly deems such conformations as energetically impractical [127].

Data for the molecule **c20n** is absent due to its exceedingly large number of rotatable bonds. A complete calculation could necessitate the review of as many as 7.6 trillion structures, a task too computationally intensive to be feasibly completed, and its search process was killed by the computer automatically due to its large calculation load. As a result, conclusive results for **c20n** were not obtained. This demonstrates that there is a limit on the size and flexibility of molecules that can be investigated by the *Diamond Energy* approach. However, **c20n** is a large molecule with a lot of flexibility, so the limit is fairly high.

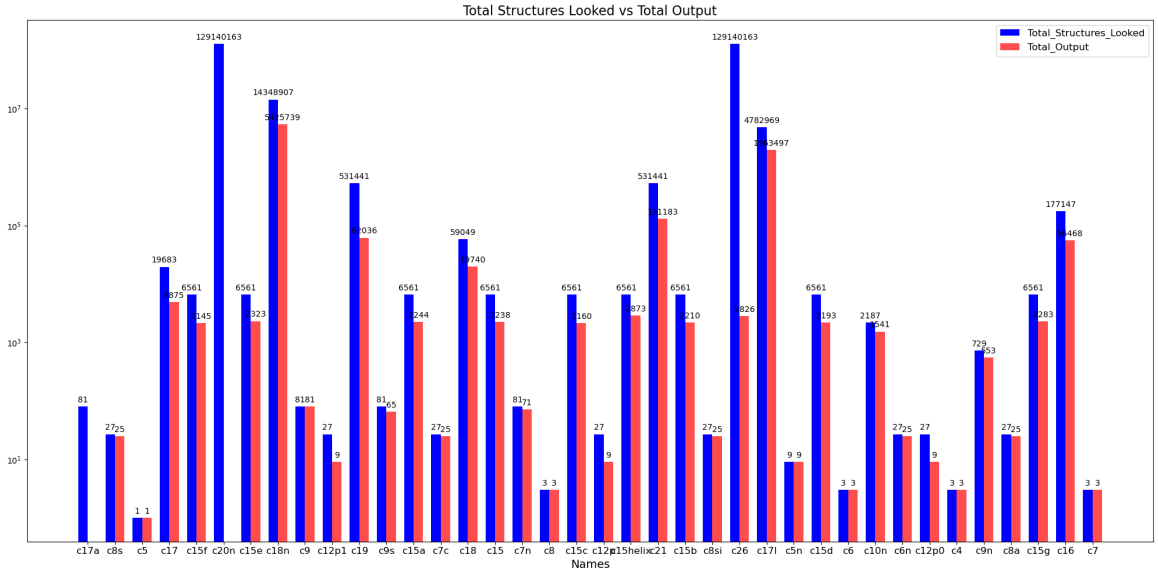


Figure 2.25 Total number of structures generated and total number of low energy conformers output by *Diamond Energy* algorithm.

The efficiency of the *Diamond Energy* algorithm is measured not just by the number of structures it assesses, but also by its computational speed. Figure 2.26 shows the total time taken for computational analysis of each test molecule. The duration of this computational time is highly variable, dependent on the number of conformers being assessed, and can range from a very small-time frame to nearly 100 million units. To accommodate these varied data points on a single graph, the y-axis uses a logarithmic scale. Equally informative are the average time metrics needed for individual structure examination and conformer generation by the *Diamond Energy* algorithm. These average times are overlaid on the graph: a yellow line represents the 'time per structure examined,' and a red line represents the 'time per structure generated,' both measured in seconds. As shown in Figure 2.26, the average time for these computations is around 10^{-3} seconds.

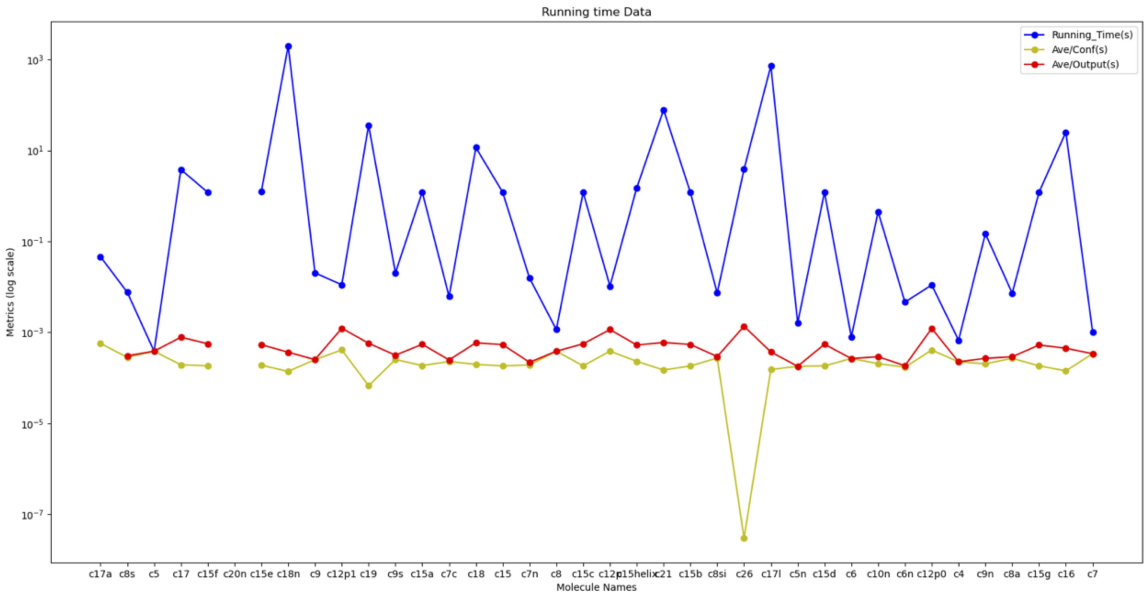


Figure 2.26 Total running time and average running time by *Diamond Energy* algorithm.

2.3.4 Evaluating *Diamond Energy*'s Conformer Search Capabilities Against MCMM, SPMC, LMCS, and MCLMCS under MM3/OPLS4

Running *Diamond Energy* in a standalone manner does not provide a complete evaluation of its ability to generate molecular conformers. A thorough assessment was carried out using a full-cycle test. In this context, *Diamond Energy* was used for generating initial structure configurations. These initial structures then went through a minimization process. The final stage of this minimization employed the Full-Matrix Newton-Raphson Technique (FMNR) and continued until the gradient values fell below 0.001 kcal/Å-mol. After minimization, vibrational frequencies were determined to confirm that the structures indeed represent true energy minima. This entire process was benchmarked against four leading conformational search algorithms: the Internal Coordinate Monte Carlo Method (MCMM), Unbounded Systematic Search (SPMC), Low-Mode Search Method (LMCS), and Monte Carlo combined with the Low-Mode Search Method (MCLMCS). All algorithms were compared using two different force fields for fairness: MM3 and OPLS4. To provide a clearer illustration of the comparison process and its associated calculations, please refer to Figure 2.27.

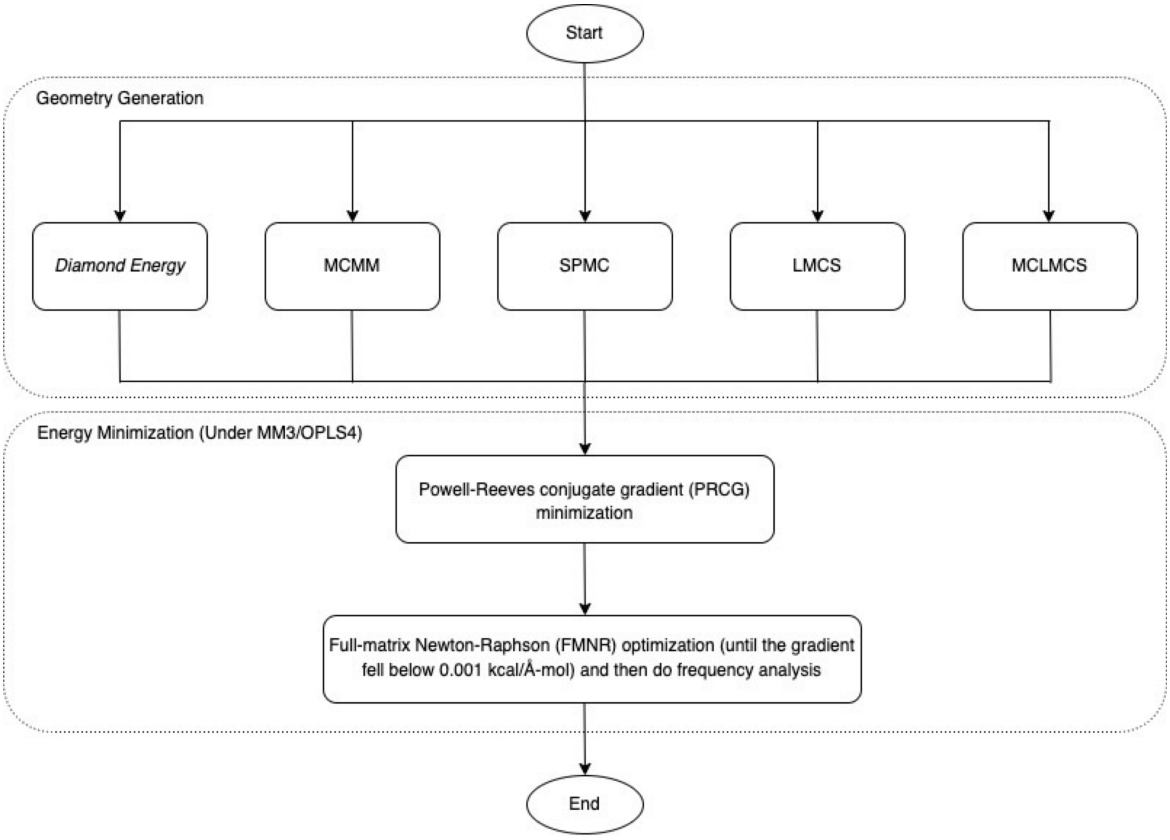


Figure 2.27 Comparative Workflow of *Diamond Energy* Algorithm and Four Leading Conformational Search Algorithms with Energy Minimization and Frequency Analysis.

2.3.4.1 An Overall Comparative Analysis of *Diamond Energy*'s Accuracy

The initial objective was to assess the accuracy of various computational methods, with a particular focus on whether *Diamond Energy* can identify the same lowest-energy conformer as other approaches. To this end, the final output files, which are frequency test log files, were parsed to extract the true minimum energies (excluding saddle points and structures that failed to converge). These minimum energies were then visualized in the Figure 2.28 (MM3).

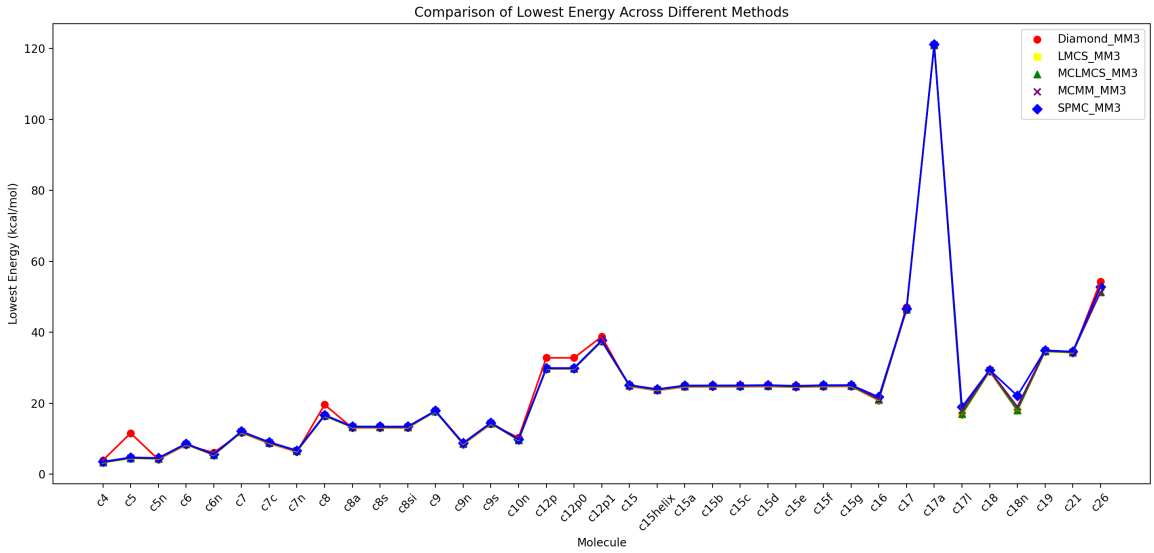


Figure 2.28 This figure shows a side-by-side comparison of the lowest energy conformers identified by various computational approaches using the MM3 force field. (The data here suggest that, for most of the test scenarios utilizing the MM3 force field, there is a consensus among the different methods regarding the identification of the conformer with the lowest energy. However, in certain cases, specifically with molecules **c5**, **c8**, **c12p**, **c12p0**, **c12p1**, and **c26**, *Diamond Energy* reported slightly higher energy values for the lowest-energy conformers. Further investigation into the molecular structures revealed that *Diamond Energy* does, in fact, recognize the same lowest-energy carbon chains as identified by the comparative methods. A detailed explanation is provided in the following paragraph to offer additional insight into this observation.

In Figure 2.28, the analysis indicates that in most test cases under the MM3 force field, all methods converge on the same lowest-energy conformer. However, exceptions were observed in specific cases, namely **c5**, **c8**, **c12p**, **c12p0**, **c12p1**, and **c26**. In these instances, *Diamond Energy* appeared to identify conformers with slightly higher energies. Upon closer examination of the structures, it was found that *Diamond Energy* does identify the same lowest-energy carbon chain as the other methods. The differences arise during the subsequent

steps, which involve adding hydrogen atoms to the carbon chain using RDKit and converting the file format for MacroModel minimization. Therefore, the minor energy differences can be not attributed to *Diamond Energy*'s inability to locate the lowest-energy structures, but rather to suboptimal hydrogen placement during intermediate processing steps, which involve the use of third-party software packages. As it stands, there is no good solution to ensure that the hydrogen atoms are added in a manner that is entirely consistent with the controls provided by other methods in MacroModel.

In addition, the structural differences found the primary variations are observed in the orientation of hydrogen atoms. To numerically quantify the differences between the conformers generated by *Diamond Energy* and other computational methods, a root-mean-square deviation (RMSD) analysis was performed. The RMSD values were computed using the RDKit package, excluding hydrogen atoms from consideration. The structure files and calculation scripts will be made available in the appendices or through a shared link listed therein. In this analysis, the LMCS conformer served as the reference standard, as it yielded results the same with those from other methods on these molecular structures. Notably, the RMSD values indicate a high degree of similarity between the structures, with the majority of the values falling below 2 Å (as shown in Figure 2.29).

The influence of hydrogen addition on the final minimization outcomes was previously demonstrated and highlighted, underscoring the complexity of the energy landscape of the test molecules. The orientation of hydrogen atoms can lead to the convergence towards different potential energy surface wells. Consequently, the study has incorporated an additional procedure with the MacroModel's applyhtreat approach for handling hydrogen. Thus, the test set comparisons now encompass six pipelines: *Diamond Energy* with RDKit for hydrogen addition followed by MM3 minimization, *Diamond Energy* with MacroModel for hydrogen addition followed by MM3 minimization, MCMM-MM3, SPMC-MM3, LMCS-MM3, and MCLMCS-MM3. The methodology for hydrogen addition using RDKit and MacroModel has been presented in a previous section 2.3.2. The results of this comprehensive analysis are presented below.

Figure 2.30 presents a comparative visualization of the lowest energy conformers obtained by employing two distinct hydrogen handling methods for *Diamond Energy* conformer search algorithm before optimization using the MacroModel software. The methods compared are the RDKit AddHs function and the MacroModel's applyhtreat function for .mae structure files, with both utilizing the MM3 force field. The illustration reveals that MacroModel's

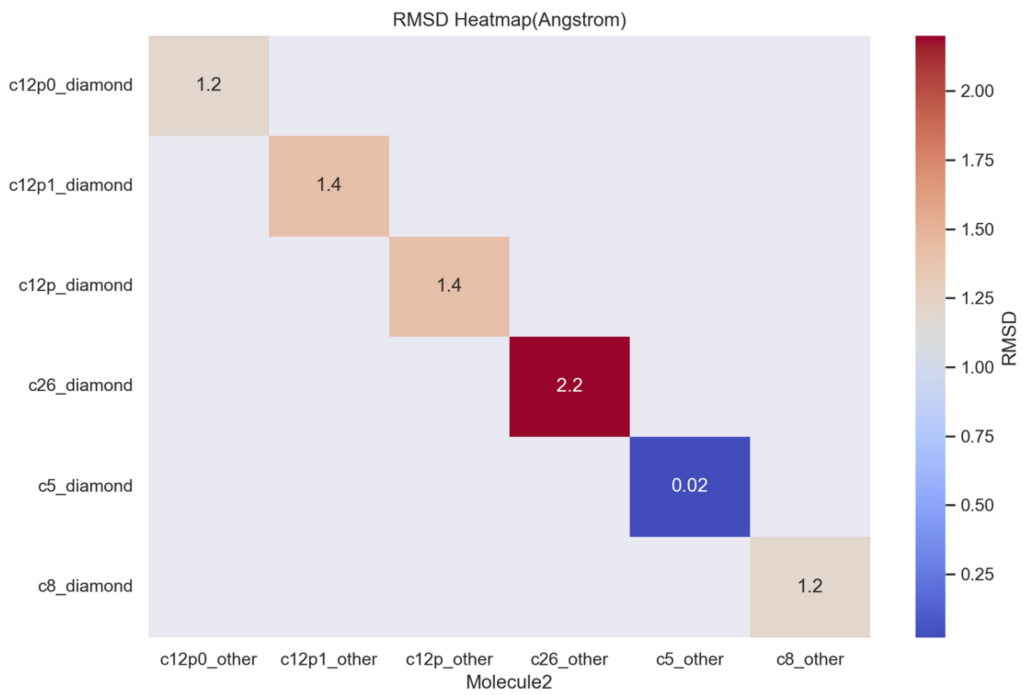


Figure 2.29 RMSD Values Comparing the Lowest-Energy Conformer from *Diamond Energy* and Other Methods in MM3 Force Field.

optimization generally enhances the convergence outcomes, indicating that the method of hydrogen addition can be instrumental in achieving more energetically favorable structures.

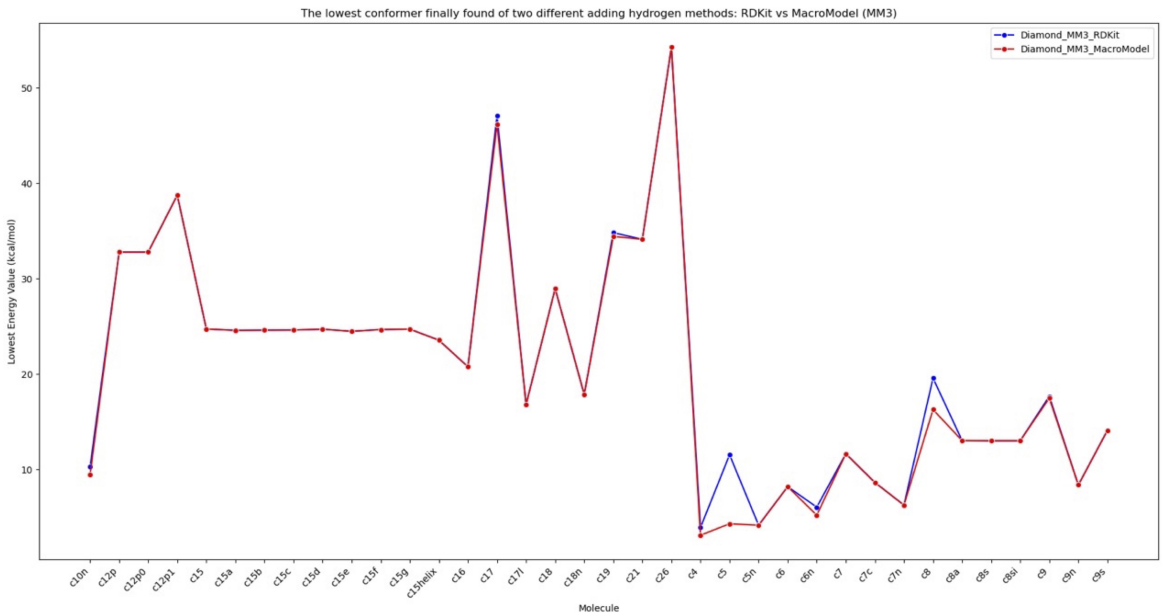


Figure 2.30 Comparative Analysis of Lowest Energy Conformers Using Distinct Hydrogen Treatment Techniques of *Diamond Energy* Prior to Optimization with MacroModel.

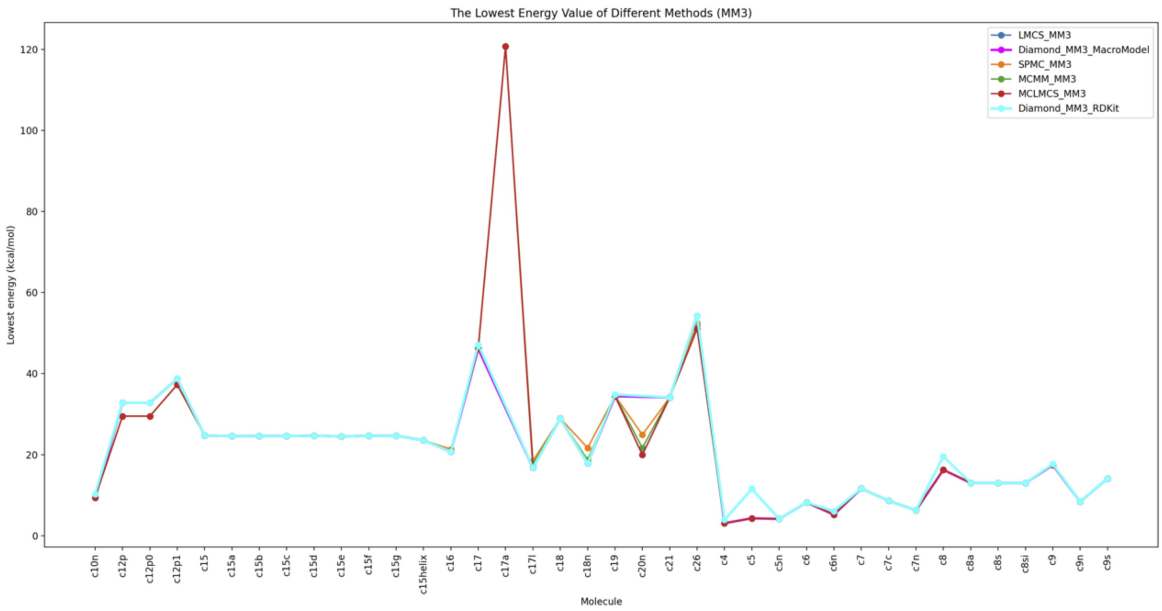


Figure 2.31 This figure shows a side-by-side comparison of the lowest energy conformers identified by various computational approaches using the MM3 force field.

Table 2.19 displays the lowest energy values identified by each of the six different pipelines: *Diamond Energy* – using the RDKit package for post-hydrogen addition before minimization,

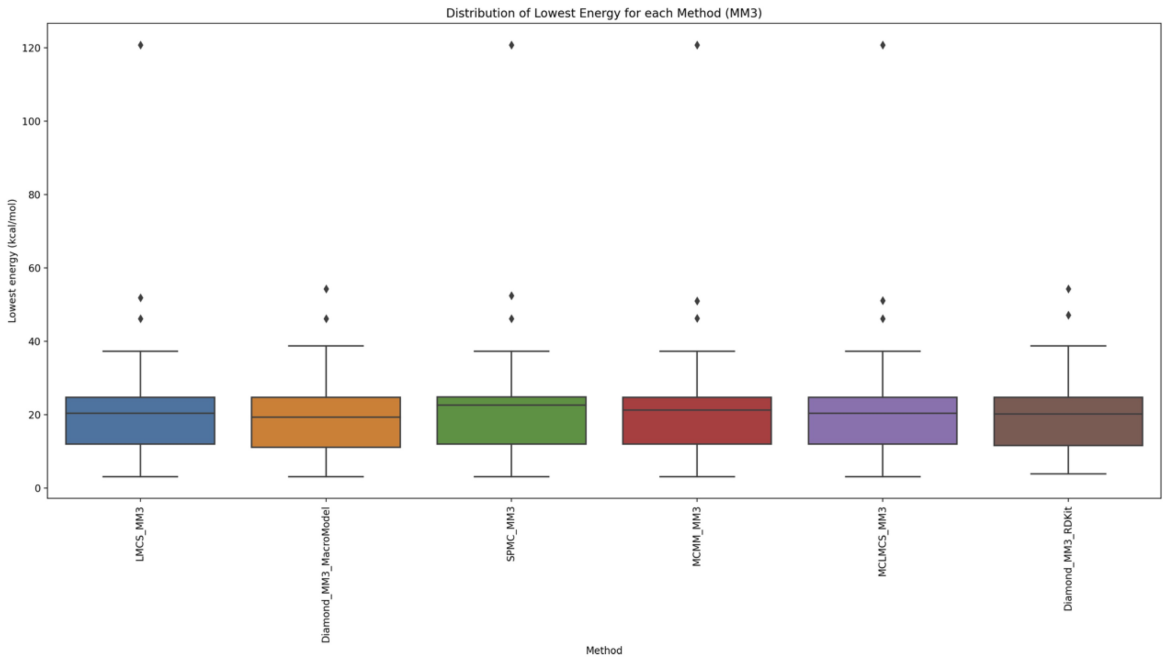


Figure 2.32 The lowest energy levels box plot compare across different methods (MM3).

Diamond Energy – employing the MacroModel function for post-hydrogen addition before minimization, and the SPMC, MCMM, MCLMCS, LMCS methods. All analyses were conducted using MM3 force field calculations. It can be observed that in most cases, *Diamond Energy* is capable of identifying the lowest energy values, similar to the other methods. In these test cases, the MacroModel function for adding hydrogen proved to be more effective than the RDKit package in some instances.

To visualize the results, Figure 2.31 illustrates a side-by-side comparison of the lowest energy conformers derived from six distinct methodologies: *Diamond Energy* combined with RDKit for hydrogen addition followed by MM3 optimization, *Diamond Energy* in conjunction with MacroModel for hydrogen addition succeeded by MM3 optimization, MCMM-MM3, SPMC-MM3, LMCS-MM3, and MCLMCS-MM3. In contrast to the previous graph, the data suggest that the MacroModel's applyhtreat function for hydrogen management in the *Diamond Energy* output aligns closely with the leading method across all test cases. This underscores the capability of the *Diamond Energy* method to map the low energy conformational landscape as effectively as established techniques. Furthermore, the results imply that the strategy for handling hydrogen prior to subsequent optimization warrants careful consideration, as it does influence the outcome.

Table 2.19 The table presents the lowest energy values calculated using the MM3 method (in kcal/mol), as identified by various techniques. These include *Diamond Energy* conformer generation followed by RDKit hydrogen post-processing and then minimization, *Diamond Energy* conformer generation followed by MacroModel hydrogen post-processing and then minimization, as well as results from the SPMC, MCMM, MCLMCS, and LMCS methods.

Molecule	Diamond_RDKit	Diamond_MacroModel	SPMC	MCMM	MCLMCS	LMCS
c4	3.90	3.09	3.09	3.09	3.09	3.09
c5	11.53	4.30	4.30	4.30	4.30	4.30
c5n	4.14	4.14	4.14	4.14	4.14	4.14
c6	8.18	8.18	8.18	8.18	8.18	8.18
c6n	6.04	5.20	5.20	5.20	5.20	5.20
c7	11.63	11.63	11.63	11.63	11.63	11.63
c7c	8.62	8.62	8.62	8.62	8.62	8.62
c7n	6.25	6.25	6.25	6.25	6.25	6.25
c8	19.57	16.27	16.26	16.26	16.26	16.26
c8a	13.02	13.02	13.02	13.02	13.02	13.02
c8s	13.00	13.00	13.00	13.00	13.00	13.00
c8si	13.00	13.00	13.00	13.00	13.00	13.00
c9	17.66	17.46	17.46	17.46	17.46	17.46
c9n	8.36	8.36	8.36	8.36	8.36	8.36
c9s	14.09	14.09	14.09	14.09	14.09	14.09
c10n	10.25	9.41	9.41	9.41	9.41	9.41
c12p	32.78	32.78	29.50	29.50	29.50	29.50
c12p0	32.78	32.78	29.50	29.50	29.50	29.50
c12p1	38.75	38.75	37.28	37.28	37.28	37.28
c15	24.71	24.71	24.71	24.71	24.71	24.71
c15a	24.57	24.57	24.57	24.57	24.57	24.57
c15b	24.60	24.60	24.60	24.60	24.60	24.60
c15c	24.61	24.61	24.61	24.61	24.61	24.61
c15d	24.69	24.69	24.69	24.69	24.69	24.69
c15e	24.48	24.48	24.48	24.48	24.48	24.48
c15f	24.66	24.66	24.66	24.66	24.66	24.66
c15g	24.70	24.70	24.70	24.70	24.70	24.70
c15helix	23.54	23.54	23.54	23.54	23.54	23.54
c16	20.76	20.76	21.34	20.84	20.76	20.76
c17	47.11	46.18	46.18	46.27	46.18	46.18
c17a	NaN	NaN	120.74	120.74	120.74	120.74
c17l	16.79	16.79	18.49	17.66	16.79	16.79
c18	28.95	28.95	28.95	28.95	28.95	28.95
c18n	17.84	17.84	21.67	18.68	17.84	17.84
c19	34.82	34.43	34.51	34.51	34.43	34.43
c20n	NaN	NaN	24.92	21.66	19.95	19.95
c21	34.12	34.12	34.12	34.17	34.12	34.12
c26	54.27	54.27	52.43	51.06	51.08	51.86

A boxplot is drawn to show how the lowest energy levels compare across different methods (Figure 2.32). This boxplot helps to see how the energy levels are spread out for each method.

The boxplot has a box that shows where the middle half of the data is, with a line in the middle that shows the median. The "whiskers" are lines that stretch out from the box to the highest and lowest values that are not considered unusual. Points that are far from the rest are marked as outliers.

Looking at the boxplot (Figure 2.32), we can see that: I. The median is the middle value of all the energy levels. If a method's median is low, it means it usually finds lower energy levels. The Diamond_MM3_MacroModel method has a low median, which means it often finds lower energy levels. II. The height of the box is the interquartile range (IQR), which shows where the bulk of the values lie. A smaller box means most of the energy levels are close together. The Diamond_MM3_MacroModel method's box is small, so its energy levels do not vary too much. III. Outliers are the points that are far away from the rest. If there aren't many outliers, the method might be more consistent. The Diamond_MM3_MacroModel and Diamond_MM3_RDKit methods do not have many outliers. IV. The whiskers show the range of the regular values. Shorter whiskers mean less variation in the energy levels. The Diamond_MM3_MacroModel method has short whiskers, so its energy levels are pretty consistent. From all this, it looks like the Diamond_MM3_MacroModel method is the best out of the six, finding lower energy levels more often.

Similarly, in calculations using the OPLS4 force field, the *Diamond Energy* appears to consistently identify lower-energy structures in most test cases as other methods did, as depicted in Table 2.20. An exception is found for the molecule **c17a**, for which data is absent. This absence is attributed to *Diamond Energy*'s assessment that the energy level for **c17a** is too high for stable existence. **c20n**, for which the data is also missed due to the limit of *Diamond Energy*, this observation is in alignment with findings from a previous study [127].

To visualize these results obtained using OPLS4, a line chart is drawn to showcase and compare the outcomes of different methods in finding the lowest energy value for various test cases. The x-axis represents the test molecules, and the y-axis indicates their energy values. Different lines represent the results from various methods, including *Diamond Energy* – RDKit with added hydrogen pipeline, *Diamond Energy* - MacroModel with added hydrogen pipeline, SPMC, LMCS, MCMM, and MCLMCS. All are calculated under the OPLS4 force field. From Figure 2.33, it can be observed that the trend of Diamond_MacroModel_OPLS4 largely overlaps with those of other methods. Meanwhile, Diamond_RDKit_OPLS4 does not

Table 2.20 The table presents the lowest energy values calculated using the OPLS4 method (in kcal/mol), as identified by various techniques. These include *Diamond Energy* conformer generation followed by RDKit hydrogen post-processing and then minimization, *Diamond Energy* conformer generation followed by MacroModel hydrogen post-processing and then minimization, as well as results from the SPMC, MCMM, MCLMCS, and LMCS methods.

Molecule	Diamond_MacroModel	Diamond_RDKit	LMCS	MCLMCS	MCMM	SPMC
c4	2.79	3.27	2.79	2.79	2.79	2.79
c5	−10.93	−4.41	−10.93	−10.93	−10.93	−10.93
c5n	3.44	3.44	3.44	3.44	3.44	3.44
c6	−1.26	−1.26	−1.26	−1.26	−1.26	−1.26
c6n	4.05	4.54	4.05	4.05	4.05	4.05
c7	−4.85	−4.85	−4.85	−4.85	−4.85	−4.85
c10n	6.55	7.04	6.55	6.55	6.55	6.55
c12p	2.56	2.56	−0.59	−0.59	−0.59	−0.59
c12p0	2.56	2.56	−0.59	−0.59	−0.59	−0.59
c12p1	12.50	12.49	7.84	7.84	7.84	7.84
c15	4.12	4.12	4.12	4.12	4.12	4.12
c15a	3.77	3.77	3.77	3.77	3.77	3.77
c15b	3.95	3.95	3.95	3.95	3.95	3.95
c15c	4.03	4.03	4.03	4.03	4.03	4.03
c15d	3.92	3.92	3.92	3.92	3.92	3.92
c15e	3.64	3.64	3.64	3.64	3.64	3.64
c15f	3.69	3.69	3.69	3.69	3.69	3.69
c15g	3.93	3.93	3.93	3.93	3.93	3.93
c15helix	2.02	2.02	2.02	2.02	2.02	2.02
c16	6.76	6.76	6.77	6.77	6.94	7.16
c17	29.62	30.01	29.63	29.63	29.82	29.63
c17l	10.94	10.46	10.63	10.95	11.20	11.44
c18	2.10	2.10	2.10	2.10	2.10	2.10
c18n	11.57	10.54	11.05	11.55	11.42	12.24
c19	14.14	14.14	13.96	14.01	13.96	14.12
c21	1.78	1.78	1.78	1.78	1.84	1.84
c26	25.18	25.18	21.08	20.82	21.26	21.04
c7c	2.91	2.91	2.91	2.91	2.91	2.91
c7n	4.67	4.67	4.67	4.67	4.67	4.67
c8	−5.24	−2.11	−5.24	−5.24	−5.24	−5.24
c8a	3.97	3.97	3.97	3.97	3.97	3.97
c8s	3.80	3.80	3.80	3.80	3.80	3.80
c8si	3.80	3.80	3.80	3.80	3.80	3.80
c9	8.92	9.11	8.92	8.92	8.92	8.92
c9n	5.92	5.92	5.92	5.92	5.92	5.92
c9s	4.48	4.48	4.48	4.48	4.48	4.48
c17a	NaN	NaN	116.23	116.23	116.23	116.23
c20n	NaN	NaN	11.39	12.07	12.55	16.25

always align with the other methods, but the differences in the values of the test molecules are not significantly large.

To further quantify and analyze the results obtained from OPLS4 calculations, a statistical box plot was drawn. As shown in Figure 2.34, the box itself represents the interquartile range (IQR), which is the range between the first and third quartiles. The line within the box indicates the median value. The 'whiskers' extend to the smallest and largest values within 1.5 times the IQR from the quartiles, and the dots outside of the whiskers represent outliers. Diamond_MacroModel_OPLS4 has the lowest median energy value, suggesting that it may be the best at consistently finding the lowest energy values. The IQR values indicate the spread of the middle 50% of the data; lower IQR values suggest a method is more consistent. The IQRs for these methods are as follows: Diamond_MacroModel_OPLS4 (3.35), Diamond_RDKit_OPLS4 (3.31), LMCS (3.46), MCLMCS (3.46), MCMM (3.46), and SPMC (3.46). The number of outliers can indicate the variability of the results; fewer outliers suggest better consistency. The counts of outliers for these methods are Diamond_MacroModel_OPLS4 (8), Diamond_RDKit_OPLS4 (6), LMCS (6), MCLMCS (7), MCMM (7), and SPMC (8). Overall, from these data, it is evident that both *Diamond Energy* pipelines are capable of identifying the lowest energy values, and they are among the four leading algorithms, namely SPMC, LMCS, MCLMCS, and MCMM. But overall, there is no big gap among all these methods.

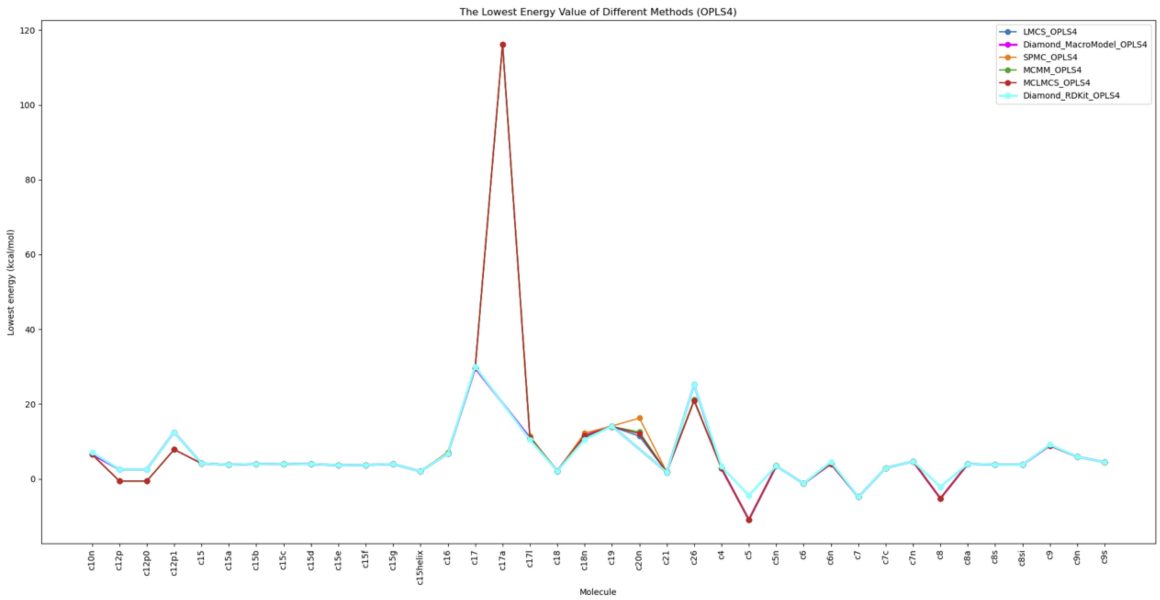


Figure 2.33 Comparison of the Minimum Energy Values Obtained Through Different Computational Methods Using the OPLS4 Force Field.

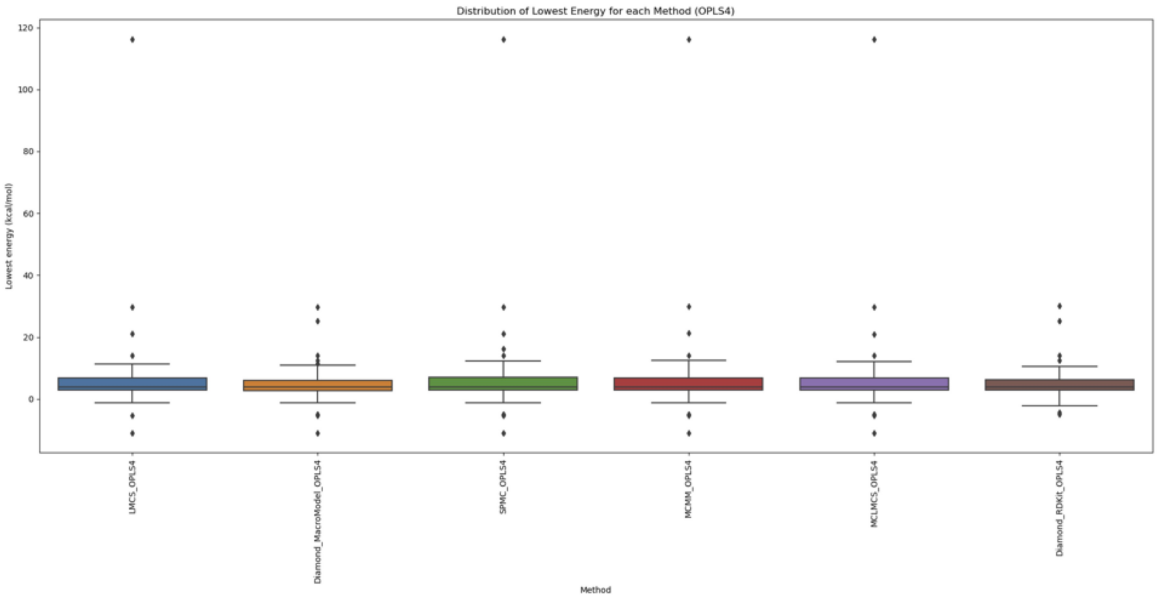


Figure 2.34 The lowest energy levels box plot compare across different methods (OPLS4).

In conclusion, based on the lowest energy identified for the entire test set from both MM3 and OPLS4 calculations, the statistical analysis shows that *Diamond Energy* is capable of identifying the lowest energy values when compared with the four leading algorithms. Post-processing steps such as adding hydrogens might also slightly affect the final results. MacroModel's function better aligns with the four leading algorithms (SPMC, LMCS, MCLMCS, and MCMM) than RDKit pipeline in these test cases used.

2.3.4.2 Completeness in Conformer Sampling by *Diamond Energy*

The following figure illustrates the total number of true minima identified by *Diamond Energy* and other computational methods. The data were extracted from the final .log files, generated after carrying out minimization and frequency tests to confirm the true minima. These tests also serve to exclude structures that have not converged or that represent saddle points.

For molecules in the range of **c16** to **c26**, the dataset for *Diamond Energy* is not complete. This is due to the necessity to truncate the number of geometries generated by *Diamond Energy* in order to optimize computational time. This limitation arises from the time-consuming process of file format conversion required to feed *Diamond Energy*'s output into MacroModel for further minimization and frequency tests. While *Diamond Energy* is

efficient in generating molecular geometries, as will be discussed later, significant time is required for file preparation to make the data compatible with MacroModel. This issue is particularly acute for molecules with long chains, such as those in the **c16** to **c26** range, which can produce up to 10 million conformers in certain cases. *Diamond Energy* produces duplicate entries for highly symmetric, linear structures like **c17l** and **c18n** due to the current conformer generation strategy. To address this issue, the search was restricted by excluding previously identified symmetric structures, effectively reducing the number of searched conformers by half. Even with these limitations, the output for molecules like **c17l** still runs into the millions. To address these computational challenges, only conformers with energies within 5 units of the lowest energy, as evaluated by *Diamond Energy*, were subjected to further minimization in MacroModel. Consequently, the data for molecules **c16** to **c26** only focus on exploring the low-energy conformer space output by *Diamond Energy*, instead of doing a complete search.

As previously noted, the method of hydrogen addition can influence the convergence toward different potential energy surface wells. To address this, the study now includes a procedure that utilizes MacroModel's approach for hydrogen handling. The test set comparisons have been expanded to feature six pipelines: *Diamond Energy* with RDKit for hydrogen addition followed by MM3 minimization, *Diamond Energy* with MacroModel for hydrogen addition followed by MM3 minimization, MCMM-MM3, SPMC-MM3, LMCS-MM3, and MCLMCS-MM3. The outcomes of this analysis are detailed below (Table 2.21). The detailed results show that both pipelines of *Diamond Energy*, the two columns counting from the left-hand side, in most cases, could identify a significantly large number of true minima, comparable to four other leading algorithms. In some cases, *Diamond Energy* even found more low-energy conformations than its counterparts. It has to be noted that, from **c16** to **c26**, the results for *Diamond Energy* are not complete due to the minimization process being costly, which prevents the full processing and minimization of all results. The number of times each pipeline identified the greatest number of true minima across all test cases is as follows. Diamond_RDKit_MM3: 24, Diamond_MacroModel_MM3: 15, LMCS_MM3: 12, SPMC_MM3: 8, MCLMCS_MM3: 8, MCMM_MM3: 8. It seems that for these test molecules, *Diamond Energy* does have some advantage over comparison methods in identifying minima.

Similarly, the same comparative analyses using the OPLS4 force field were conducted. The test set was expanded to include six pipelines: *Diamond Energy* combined with RDKit for hydrogen addition followed by OPLS4 minimization; *Diamond Energy* with MacroModel for hydrogen addition, again followed by OPLS4 minimization; and four other pipelines using

Table 2.21 The table displays the minima calculated using the MM3 method, identified through various techniques. This includes conformer generation with *Diamond Energy* followed by RDKit hydrogen post-processing and minimization, conformer generation with *Diamond Energy* followed by MacroModel hydrogen post-processing and minimization, and the results from the SPMC, MCMM, MCLMCS, and LMCS methods.

Molecule	Diamond_MacroModel	Diamond_RDKit	LMCS	MCLMCS	MCMM	SPMC
c10n	856	1537	876	888	889	891
c12p	9	9	2	2	2	2
c12p0	9	9	2	2	2	2
c12p1	9	9	7	7	7	7
c15	2223	2231	1254	1289	1276	1405
c15a	2234	2238	1276	1313	1292	1430
c15b	2195	2205	1248	1316	1281	1413
c15c	2152	2158	1280	1328	1322	1461
c15d	2183	2187	1439	1462	1475	1559
c15e	2307	2315	1108	1137	1148	1290
c15f	2132	2142	1458	1506	1509	1657
c15g	2275	2272	1310	1391	1358	1527
c15helix	2866	2695	851	870	866	925
c16	3821	14	3958	3937	3726	2228
c17	4867	51	1256	1354	1412	1014
c17l	197	5139	3826	3445	3414	1930
c18	55	52	2675	2371	1885	1537
c18n	226	6910	3792	3312	2559	2457
c19	98	16	2446	2322	1695	1458
c21	297	89	2423	2103	1458	1940
c26	2748	2783	2436	2098	1112	929
c4	3	2	3	3	3	3
c5	1	1	1	1	1	1
c5n	7	7	7	7	7	7
c6	3	3	3	3	3	3
c6n	25	24	20	20	20	20
c7	3	3	1	1	1	1
c7c	25	25	33	33	33	33
c7n	37	69	52	52	52	52
c8	1	1	1	1	1	1
c8a	25	25	35	35	35	35
c8s	25	25	23	23	23	23
c8si	25	25	23	23	23	24
c9	69	72	12	11	10	12
c9n	301	547	366	366	366	366
c9s	65	65	96	96	96	96

Table 2.22 The table displays the minima calculated using the OPLS4 method, identified through various techniques. This includes conformer generation with *Diamond Energy* followed by RDKit hydrogen post-processing and minimization, conformer generation with *Diamond Energy* followed by MacroModel hydrogen post-processing and minimization, and the results from the SPMC, MCMM, MCLMCS, and LMCS methods.

Molecule	Diamond_MacroModel	Diamond_RDKit	LMCS	MCLMCS	MCMM	SPMC
c10n	856	855	1371	1403	1414	1421
c12p	9	9	2	2	2	2
c12p0	9	9	2	2	2	2
c12p1	9	9	7	7	7	7
c15	2233	2236	1679	1788	1813	1948
c15a	2239	2240	1756	1811	1839	1842
c15b	2203	2205	1647	1754	1724	1903
c15c	2154	2156	1827	1875	1856	1923
c15d	2187	2191	1895	1994	1947	1940
c15e	2316	2320	1522	1637	1658	1807
c15f	2137	2141	1644	1741	1683	1941
c15g	2272	2276	1776	1906	1903	2114
c15helix	2864	2869	1505	1552	1577	1686
c16	3820	14	5020	5189	5397	2522
c17	4864	51	1549	1673	1745	980
c17l	197	5126	5067	5550	6091	2341
c18	55	52	3724	3176	2756	1945
c18n	226	6900	4642	5265	5315	2367
c19	98	80	3048	2845	1963	998
c21	299	1423	3210	2934	2064	1209
c26	2761	2786	2536	2167	1264	1392
c4	3	2	3	3	3	3
c5	1	1	1	1	1	1
c5n	7	7	7	7	7	7
c6	3	3	3	3	3	3
c6n	25	24	22	22	22	22
c7	3	3	1	1	1	1
c7c	25	25	37	37	37	37
c7n	37	69	58	58	58	58
c8	1	1	1	1	1	1
c8a	25	25	37	37	37	37
c8s	25	25	25	25	25	25
c8si	25	25	25	25	25	25
c9	58	66	12	12	11	11
c9n	301	301	511	513	512	513
c9s	65	65	115	115	115	115

Table 2.23 Conformers within a 3 kcal/mol range of the lowest energy, identified by various pipelines using MM3 calculations, include those generated with *Diamond Energy* followed by RDKit hydrogen post-processing and minimization, *Diamond Energy* followed by MacroModel hydrogen post-processing and minimization, along with results from the SPMC, MCMM, MCLMCS, and LMCS methods.

Molecule	Diamond_MacroModel	Diamond_RDKit	LMCS	MCLMCS	MCMM	SPMC
c10n	209	475	142	142	142	142
c12p	9	9	1	1	1	1
c12p0	9	9	1	1	1	1
c12p1	9	9	7	7	7	7
c15	135	131	172	171	167	174
c15a	141	143	164	163	159	165
c15b	111	106	152	151	147	152
c15c	115	115	160	160	156	160
c15d	150	151	178	179	178	177
c15e	126	125	141	141	139	141
c15f	115	115	176	172	174	177
c15g	158	155	180	180	175	185
c15helix	102	90	96	91	91	91
c16	1377	10	977	833	621	283
c17	429	42	259	258	283	197
c17l	76	881	572	337	245	256
c18	55	52	395	370	276	266
c18n	82	1029	622	342	140	529
c19	82	16	358	337	222	214
c21	127	89	285	270	185	223
c26	118	118	715	454	146	209
c4	3	3	3	3	3	3
c5	1	1	1	1	1	1
c5n	7	7	5	5	5	5
c6	3	3	3	3	3	3
c6n	17	21	12	12	12	12
c7	3	3	1	1	1	1
c7c	17	17	22	22	22	22
c7n	28	39	22	22	22	22
c8	3	3	1	1	1	1
c8a	18	17	20	20	20	20
c8s	17	19	12	12	12	12
c8si	17	19	12	12	12	12
c9	57	66	8	7	8	7
c9n	116	151	79	79	79	79
c9s	31	32	36	36	36	36

Table 2.24 Conformers within a 3 kcal/mol range of the lowest energy, identified by various pipelines using OPLS4 calculations, include those generated with *Diamond Energy* followed by RDKit hydrogen post-processing and minimization, *Diamond Energy* followed by MacroModel hydrogen post-processing and minimization, along with results from the SPMC, MCMM, MCLMCS, and LMCS methods.

Molecule	Diamond_MacroModel	Diamond_RDKit	LMCS	MCLMCS	MCMM	SPMC
c10n	458	622	351	351	351	351
c12p	9	9	1	1	1	1
c12p0	9	9	1	1	1	1
c12p1	9	9	6	6	6	6
c15	191	188	306	314	295	318
c15a	142	144	193	196	185	190
c15b	161	162	254	249	246	262
c15c	189	187	304	304	294	303
c15d	208	208	265	273	258	266
c15e	116	110	138	140	135	142
c15f	128	129	204	200	194	206
c15g	191	186	252	255	248	263
c15helix	198	195	185	186	187	189
c16	2048	13	1449	1301	1220	873
c17	656	51	483	478	542	257
c17l	161	3158	1707	1662	1686	713
c18	55	52	859	576	486	355
c18n	177	2610	1414	1344	808	589
c19	86	80	536	459	251	159
c21	223	583	532	489	316	137
c26	236	198	598	359	172	274
c4	3	3	3	3	3	3
c5	1	1	1	1	1	1
c5n	9	9	7	7	7	7
c6	3	3	3	3	3	3
c6n	25	25	18	18	18	18
c7	2	3	1	1	1	1
c7c	23	23	29	29	29	29
c7n	35	55	34	34	34	34
c8	3	3	1	1	1	1
c8a	22	22	22	22	22	22
c8s	22	22	16	16	16	16
c8si	22	22	16	16	16	16
c9	33	54	6	6	6	6
c9n	193	193	152	152	152	152
c9s	39	40	50	50	50	50

Table 2.25 Conformers within a 5 kcal/mol range of the lowest energy, identified by various pipelines using MM3 calculations, include those generated with *Diamond Energy* followed by RDKit hydrogen post-processing and minimization, *Diamond Energy* followed by MacroModel hydrogen post-processing and minimization, along with results from the SPMC, MCMM, MCLMCS, and LMCS methods.

Molecule	Diamond_MacroModel	Diamond_RDKit	LMCS	MCLMCS	MCMM	SPMC
c10n	594	1150	544	551	550	551
c12p	9	9	2	2	2	2
c12p0	9	9	2	2	2	2
c12p1	9	9	7	7	7	7
c15	473	467	728	738	720	763
c15a	508	501	735	741	722	765
c15b	444	443	722	737	718	758
c15c	457	455	744	754	726	786
c15d	519	526	788	777	781	802
c15e	499	498	675	691	695	730
c15f	458	460	807	798	803	853
c15g	534	530	756	768	754	822
c15helix	432	395	419	433	427	436
c16	2604	13	3150	2917	2549	1568
c17	1083	51	793	810	826	578
c17l	133	2873	2830	2260	2124	1229
c18	55	52	1598	1388	982	899
c18n	145	3481	2874	2157	1441	1762
c19	98	16	1609	1475	974	871
c21	303	89	1392	1193	759	1042
c26	460	442	1954	1584	676	601
c4	3	3	3	3	3	3
c5	1	1	1	1	1	1
c5n	9	9	7	7	7	8
c6	3	3	3	3	3	3
c6n	25	25	20	20	20	20
c7	3	3	1	1	1	1
c7c	24	24	32	32	32	32
c7n	39	67	48	48	48	48
c8	3	3	1	1	1	1
c8a	23	23	27	27	27	27
c8s	22	22	20	20	20	20
c8si	22	22	20	20	20	21
c9	57	54	10	9	8	10
c9n	257	389	256	256	256	256
c9s	50	51	72	72	72	72

Table 2.26 Conformers within a 5 kcal/mol range of the lowest energy, identified by various pipelines using OPLS4 calculations, include those generated with *Diamond Energy* followed by RDKit hydrogen post-processing and minimization, *Diamond Energy* followed by MacroModel hydrogen post-processing and minimization, along with results from the SPMC, MCMM, MCLMCS, and LMCS methods.

Molecule	Diamond_MacroModel	Diamond_RDKit	LMCS	MCLMCS	MCMM	SPMC
c10n	802	836	1053	1060	1062	1071
c12p	9	9	2	2	2	2
c12p0	9	9	2	2	2	2
c12p1	9	9	7	7	7	7
c15	611	608	1102	1143	1115	1236
c15a	571	569	937	947	912	957
c15b	561	556	1017	1032	1000	1106
c15c	605	609	1170	1149	1126	1172
c15d	641	645	1054	1110	1039	1042
c15e	541	524	843	835	815	909
c15f	487	491	896	886	841	954
c15g	630	639	1029	1036	1020	1163
c15helix	664	672	788	808	794	835
c16	3625	14	4044	3965	3845	2110
c17	1427	51	1155	1196	1286	673
c17l	197	5125	4348	4614	4989	1943
c18	55	52	2667	2105	1715	1155
c18n	225	6486	3855	4142	3621	1992
c19	98	80	2185	1962	1194	641
c21	303	1435	2191	1943	1299	700
c26	741	652	1963	1549	791	911
c4	3	3	3	3	3	3
c5	1	1	1	1	1	1
c5n	9	9	7	7	7	7
c6	3	3	3	3	3	3
c6n	25	25	22	22	22	22
c7	2	3	1	1	1	1
c7c	24	24	34	34	34	34
c7n	39	71	54	54	54	54
c8	3	3	1	1	1	1
c8a	25	25	33	33	33	33
c8s	24	24	23	23	23	23
c8si	24	24	23	23	23	23
c9	57	54	10	10	9	9
c9n	299	297	397	397	397	397
c9s	59	60	95	95	95	95

OPLS4: MCMM, SPMC, LMCS, and MCLMCS. The outcomes of this extended analysis are detailed below in Table 2.22. The comprehensive results indicate that, in most instances, the two *Diamond Energy* pipelines—identified in the leftmost two columns—could consistently identify a significantly larger number of true minima when compared to the four other leading algorithms. In some scenarios, *Diamond Energy* was even more successful in finding low-energy conformations than its competitors. It is important to note that for compounds **c16** to **c26**, the data for *Diamond Energy* pipelines are incomplete. The high computational costs associated with the minimization process have precluded the full processing and analysis of all potential minima. The total number of times that each pipeline has identified the most minima are as follows: Diamond_RDKit_OPLS4: 23, Diamond_MacroModel_OPLS4: 13, LMCS_OPLS4: 13, SPMC_OPLS4: 12, MCMM_OPLS4: 12, MCLMCS_OPLS4: 11. These results demonstrate that the *Diamond Energy* pipelines have an advantage in identifying minima for the test molecules.

Besides the number of true minima found, there are two other criteria that also help to evaluate the performance of these methods in exploring the low-energy space of the test molecules: the number of conformers within 3 kcal/mol and 5 kcal/mol of the lowest energy value range. These results are presented in the following sections.

The results of the conformers within 3 kcal/mol of the lowest energy range identified by various pipelines using MM3 calculations are detailed in Table 2.23. The data shows that, in most instances, the *Diamond Energy* pipelines — represented in the two leftmost columns — could identify a substantial number of conformers within 3 kcal/mol of the lowest energy range. This performance is comparable to four other leading algorithms. Notably, *Diamond Energy* sometimes identified more low-energy conformations than its counterparts. However, it should be noted that for compounds **c16** to **c26**, the *Diamond Energy* results are incomplete. The counts for how many conformers each method identified within 3 kcal/mol of the lowest energy range are as follows: Diamond_RDKit_MM3: 17, SPMC_MM3: 15, LMCS_MM3: 14, Diamond_MacroModel_MM3: 13, MCLMCS_MM3: 10, MCMM_MM3: 7. These findings suggest that the *Diamond Energy* algorithm is competent in exploring the space within 3 kcal/mol of the lowest energy range for the test molecules.

Similarly, the results of the conformers within 3 kcal/mol of the lowest energy range, as identified by various pipelines using OPLS4 calculations, are detailed in Table 2.24. The data indicate that the *Diamond Energy* pipelines — represented in the two leftmost columns — could, in most cases, identify a significant number of conformers within 3 kcal/mol of the lowest energy range. The number of conformers identified within 3 kcal/mol of

the lowest energy range by each method is as follows: Diamond_RDKit_OPLS4: 20, Diamond_MacroModel_OPLS4: 16, SPMC_OPLS4: 13, LMCS_OPLS4: 11, MCLMCS_OPLS4: 10, MCMM_OPLS4: 7.

Table 2.25 presents the results of conformers identified by various methods that are within the lowest energy value range of 5 kcal/mol, utilizing the MM3 force field. The methods compared include: *Diamond Energy* with MacroModel, where hydrogen is added before the minimization pipeline; *Diamond Energy* with RDKit, also adding hydrogen before minimization; along with LMCS, MCLMCS, MCMM, and SPMC. The number of conformers each method found within 5 kcal/mol of the lowest energy range are as follows: Diamond_RDKit_MM3: 17, SPMC_MM3: 16, Diamond_MacroModel_MM3: 14, LMCS_MM3: 13, MCLMCS_MM3: 7, MCMM_MM3: 7. These results indicate that the *Diamond Energy* algorithm is effective at exploring the conformational space within 5 kcal/mol of the lowest energy range for the test molecules.

Similarly, Table 2.26 presents the results of conformers identified by various methods that are within the lowest energy value range of 5 kcal/mol, utilizing the OPLS4 force field. The number of conformers each method found within 5 kcal/mol of the lowest energy range is as follows: SPMC_OPLS4: 17, Diamond_RDKit_OPLS4: 15, LMCS_OPLS4: 14, Diamond_MacroModel_OPLS4: 13, MCLMCS_OPLS4: 9, MCMM_OPLS4: 8. These results indicate that the *Diamond Energy* algorithm performs competitively in exploring the conformational space within 5 kcal/mol of the lowest energy range for the test molecules using an OPLS4 force field, comparable to its counterparts.

In conclusion, from the results of true minima identified, conformers found within 3 kcal/mol and 5 kcal/mol of the lowest energy range, using both MM3 and OPLS4 force field calculations, *Diamond Energy* in most test cases, its pipelines are as competent as the four leading algorithms compared: SPMC, LMCS, MCLMCS, and MCMM. It seems that this algorithm does have a good ability to explore the low-energy conformational space of the test molecules.

2.3.4.3 Computational Efficiency: Assessing the Running Speed of *Diamond Energy*

The total execution time required to generate all molecular structures for a test molecule, as well as the average time needed to produce a single structure, are compared across *Diamond Energy* (using RDKit package for adding hydrogen before minimization and four other

computational methods (Figure 2.35 and Figure 2.36). For *Diamond Energy*, time data are extracted from the 'running_record' log file. These times are calculated using Python's time package, measuring the difference between start and end times, with seconds as the unit of measurement. For the other methods in the MacroModel suite, runtimes can be obtained from their respective log files. These files provide 'User CPU Time,' which represents the cumulative time required for both geometry generation and minimization. The time specifically devoted to conformer generation is calculated by subtracting the minimization time from the total 'User CPU Time'. And it shows that *Diamond Energy* did run much faster than other strategies in generating molecular structures.

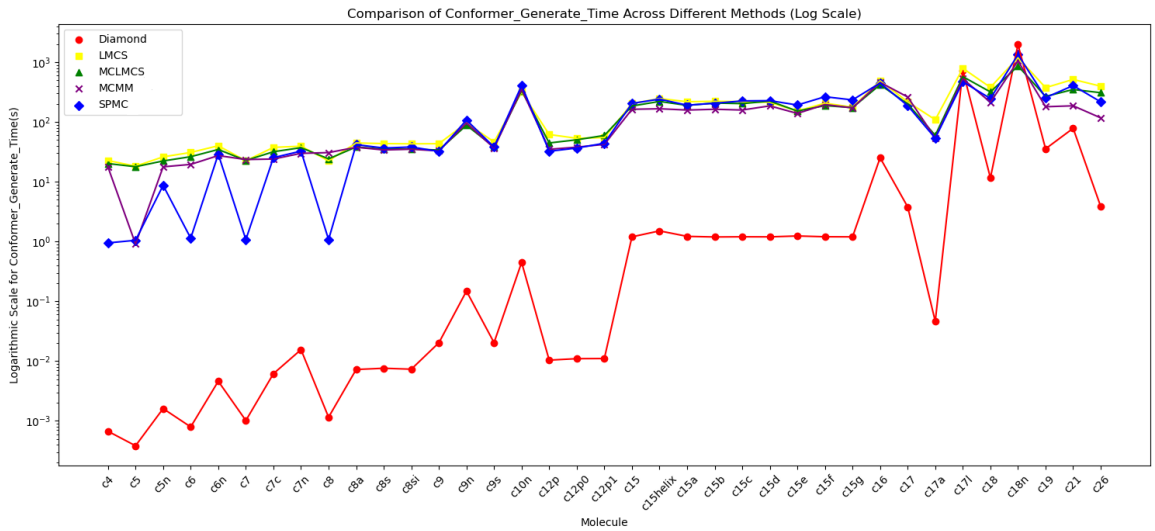


Figure 2.35 The total execution time required to generate all molecular structures for a test molecule of *Diamond Energy* and other methods.

The overall runtime and average for executing a conformer search both using MM3 and OPLS4 is presented in Figure 2.37, Figure 2.38, Figure 2.39, Figure 2.40. For *Diamond Energy*, this data is obtained by summing the running times recorded in the log files generated during conformer generation and subsequent calculations in MacroModel. For other methods which are part of MacroModel suite, the total runtime is directly extracted from MacroModel's output log files, specifically from the 'User CPU Time' entry, which excludes queue waiting and task submission times. In calculations using the MM3 force field, *Diamond Energy* appears to outperform other methods in most cases. However, in calculations involving the OPLS4 force field, *Diamond Energy* performs less efficiently in some cases than its counterparts for complete conformer searches. This might be attributed to that the structures generated by *Diamond Energy* might be not as readily optimized by the OPLS4 force field, thus requiring more time for minimization within MacroModel.

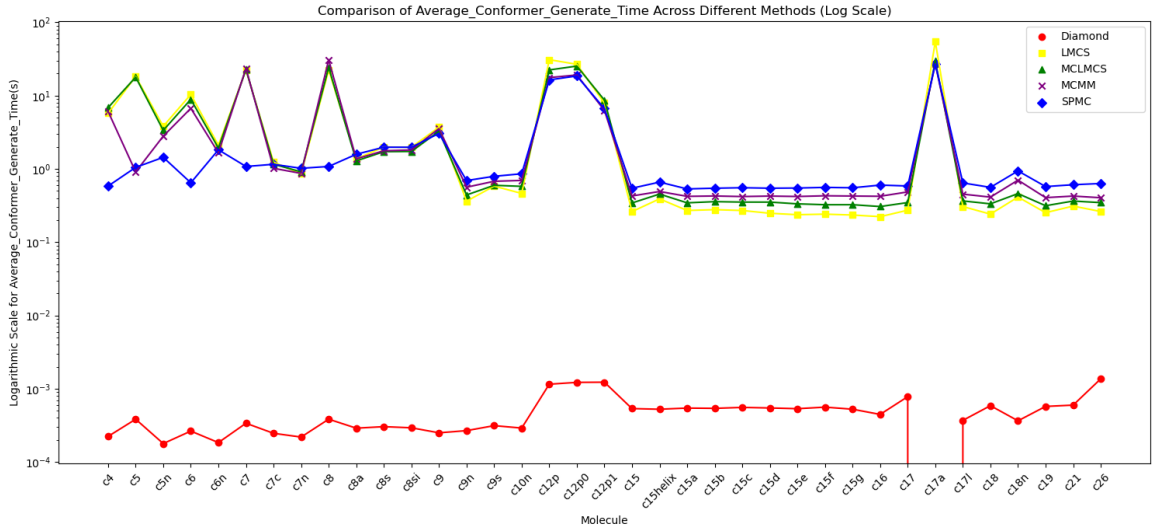


Figure 2.36 The average time needed to produce a single structure of *Diamond Energy* and other methods.

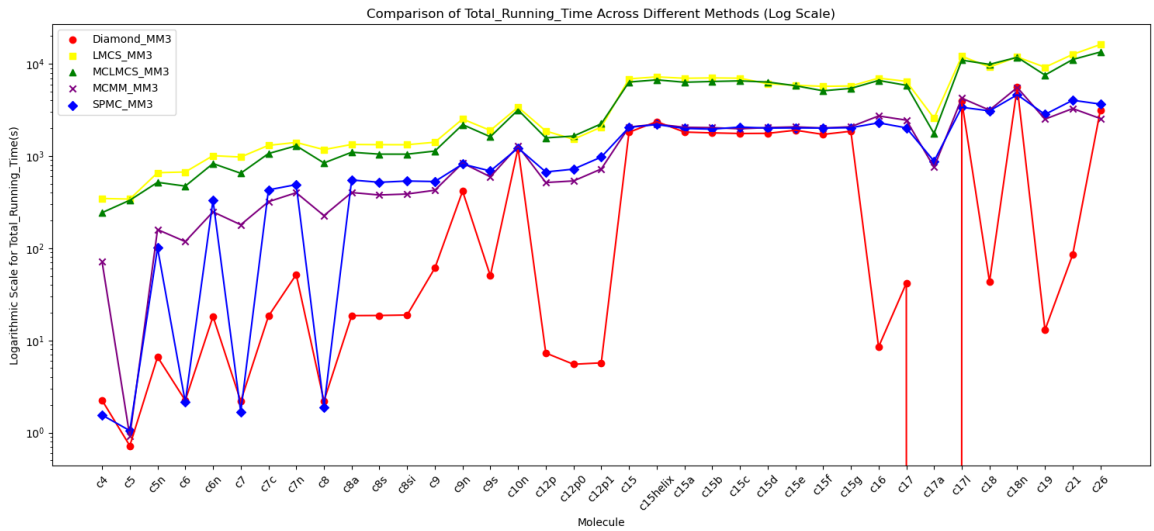


Figure 2.37 Total Execution Time for Conformer Search Comparing *Diamond Energy* and Other Methods Using the MM3 Force Field.

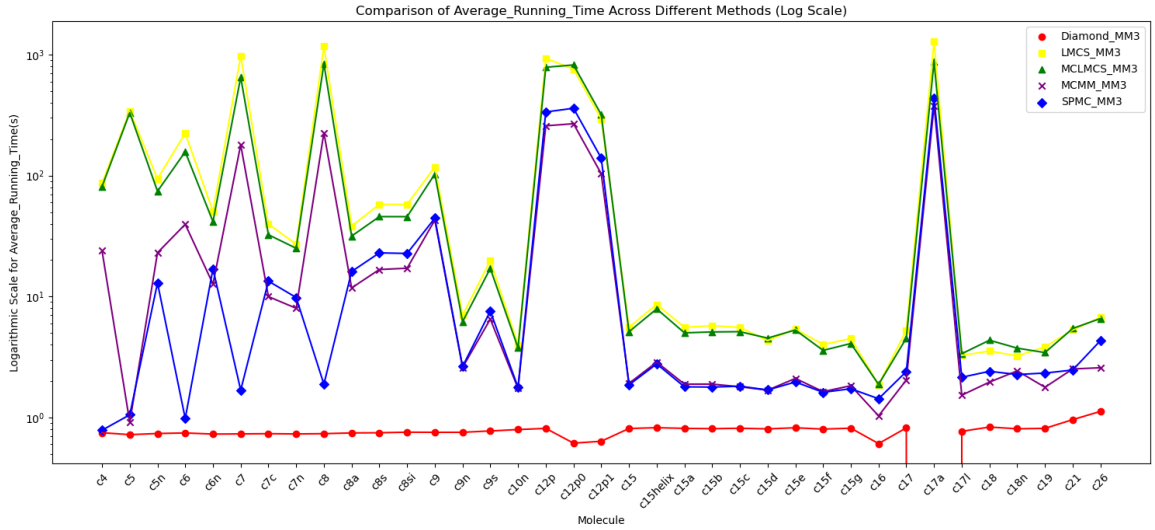


Figure 2.38 Average Execution Time for Conformer Search Using *Diamond Energy* and Other Methods with the MM3 Force Field.

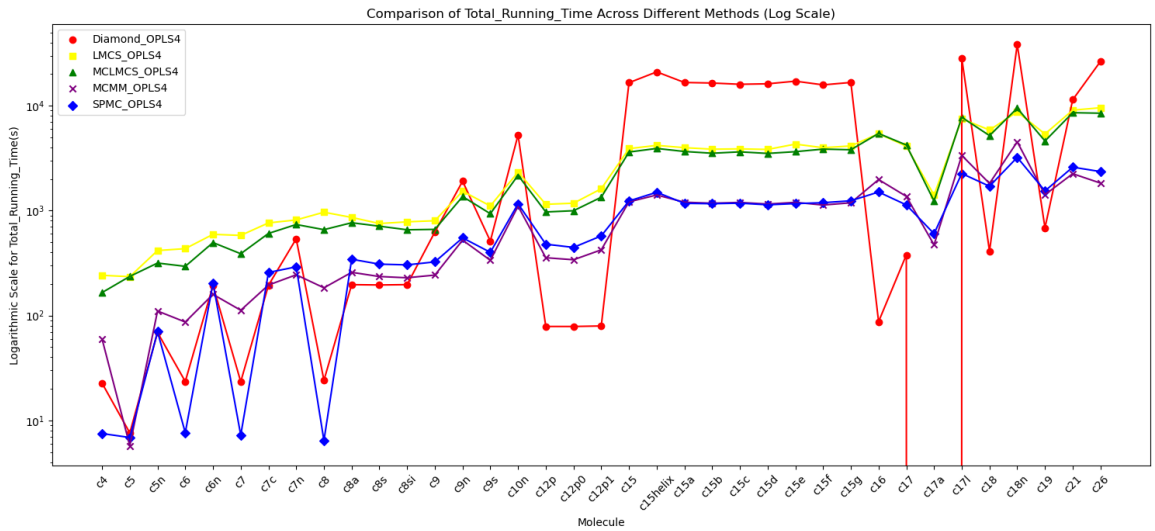


Figure 2.39 Total Execution Time for Conformer Search Comparing *Diamond Energy* and Other Methods Using the OPLS4 Force Field.

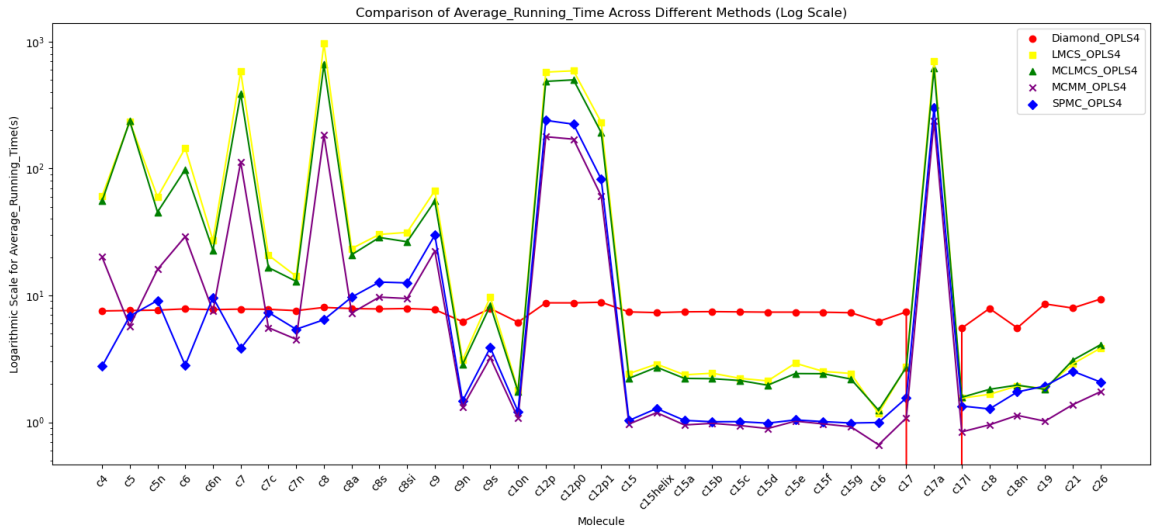


Figure 2.40 Average Execution Time for Conformer Search Using *Diamond Energy* and Other Methods using the OPLS4 Force Field.

2.3.5 Case Study: Utilizing *Diamond Energy* to Interpret Experimental Data

Aggarwal and colleagues [1] recently studied the helical tendencies of a homologous series of hydrocarbons, highlighting a distinct 'odd-even effect' in the helicity of all-syn methyl-substituted molecules. They found that hydrocarbon chains with an even number of carbon atoms tend to assume well-defined helical structures, primarily because both chain ends maintain a consistent 'gauche' (g) dihedral angle. This arrangement effectively minimizes gauche interactions, thereby enhancing structural stability. Conversely, chains with an odd number of carbon atoms are also capable of forming helical structures, but they are constrained to having one 'trans' (t) and one 'gauche' (g) dihedral angle at the termini. As a result, even-numbered chains exhibit a preference for helical configuration, since both termini contribute synergistically to a unified helical twist, often termed as the 'screw-sense.' In contrast, the inability of odd-numbered chains to sustain a consistent helical twist leads them to adopt conformations that are less regular and less predictable. To further explore these observations, the *Diamond Energy* program was employed to analyze the ensembles of several molecules (Figure 2.41), similar to those discussed in Aggarwal's paper, in order to evaluate the statistical distribution of these conformations.

In their study, they employed the MCMM method in MacroModel for conformational search, in MMFFs force field and implicit $CHCl_3$ environment. This was followed by structure

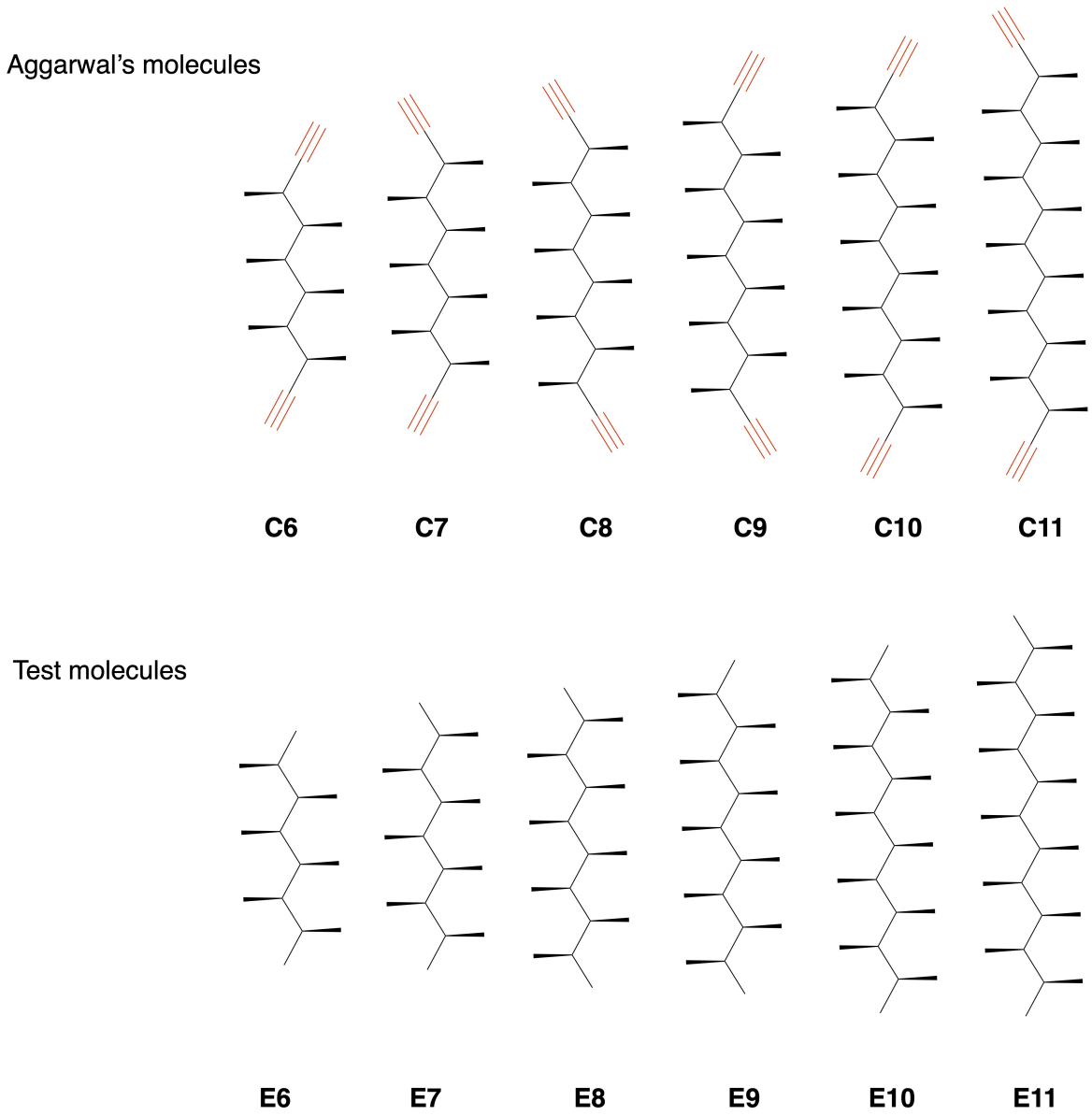


Figure 2.41 Comparative analysis of molecules investigated in Aggarwal et al.'s study (top) and molecules subjected to *Diamond Energy* evaluation (bottom).

minimization through the Truncated Newton Conjugate Gradients (TNCG). Subsequently, these conformers underwent DFT geometry optimization and frequency calculations. The researchers employed the B3LYP-D2 functional along with the 6-311G(d) basis set in a gas-phase simulation. As a result, they identified varying numbers of clustering conformers for different carbon chain lengths: 29 for **C6**, 47 for **C7**, 31 for **C8**, 97 for **C9**, 45 for **C10**, and 130 for **C11**.

The principal findings of the study are illustrated in Figure 2.42 and reveal that even-numbered, all-syn methyl-substituted hydrocarbons tend to exhibit a higher rate of helical configurations, predominantly favoring a right-handed helix. In contrast, their odd-numbered molecules have an equal number of right-handed and left-handed helical conformers.

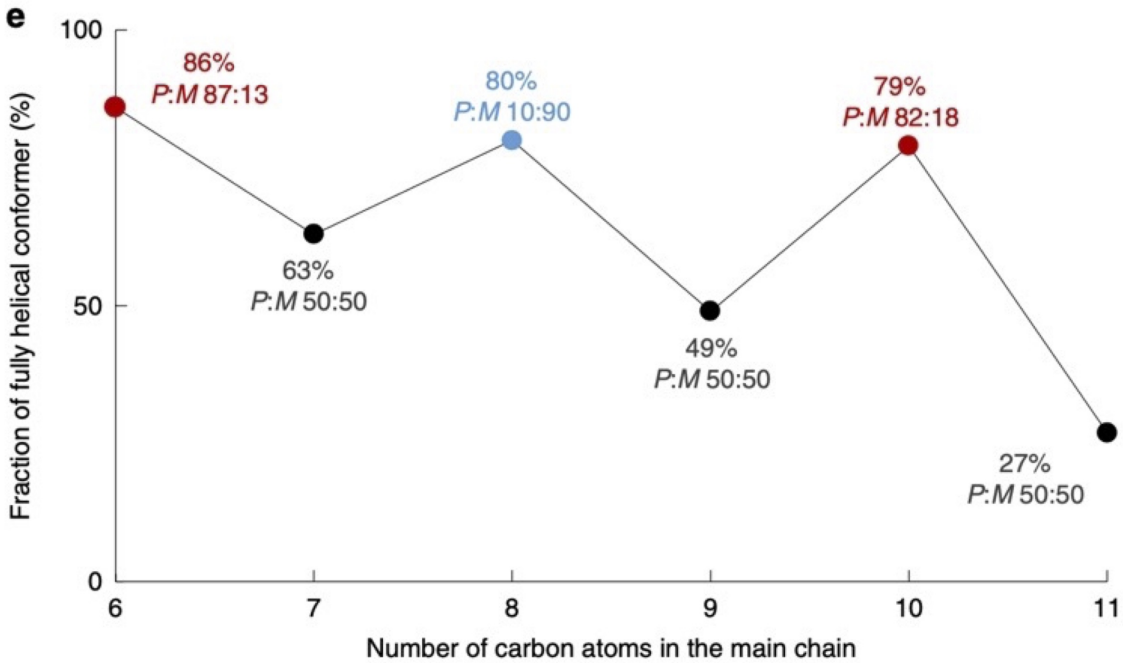


Figure 2.42 Key Findings from the Study by Aggarwal et al [1].

To explore the observations made by Aggarwal et al., a conformer search was conducted for a series of analogous hydrocarbons using *Diamond Energy*. Structure minimization was done by Powell-Reeves conjugate gradient with MacroModel, involving 2,500 iterations and a gradient convergence criterion of 0.05. Calculations were performed both under the MM3 force field in gas-phase conditions and with the MMFFs force field in an implicit $CHCl_3$ environment. The outcomes of this comparative study are presented in Figure 2.43 and Figure 2.44. The helix conformer count follows the definition in the Aggarwal paper, if the dihedral angles follow a t g– t g– pattern, then this conformer is regarded as a right-handed

helix, and if the dihedral angles follow a t g+ t g+ pattern, then this conformer is regarded as left-handed helix.

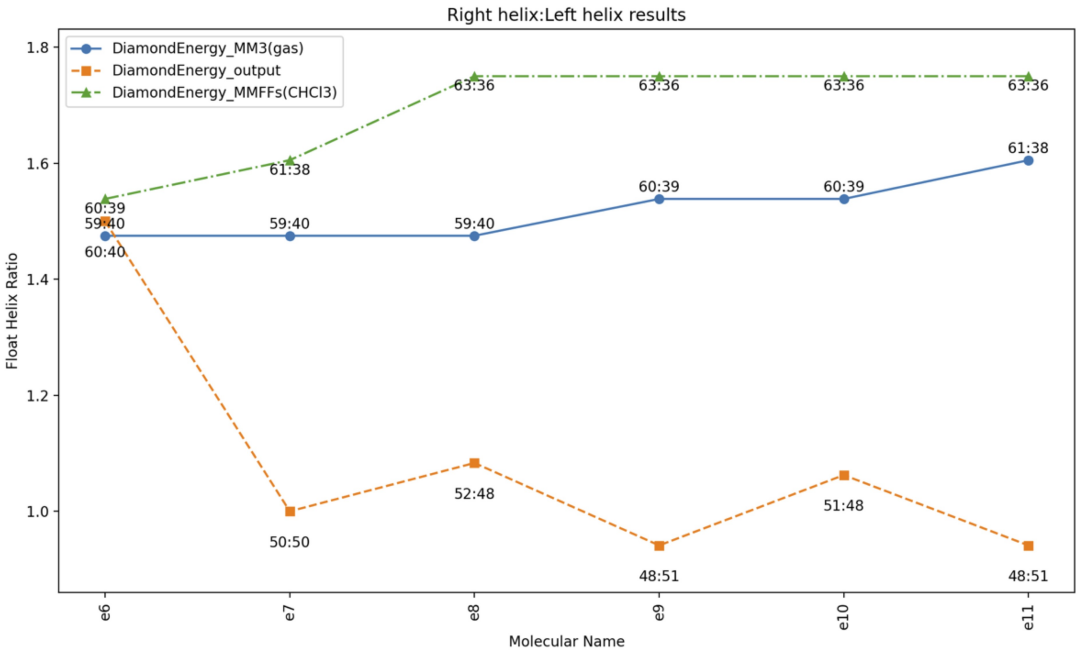


Figure 2.43 Right helix: Left helix ratio in three *Diamond Energy* conformer generation results.

Although the findings did not consistently align with prior observations that even-numbered hydrocarbons are more likely to form helix conformers than odd-numbered ones, certain patterns were still evident. For instance, these results found that the tendency to form helical structures increases with the length of the carbon chain.

In the raw data from *Diamond Energy*, before any further minimization steps, it was noted that molecules with an even number of atoms have a larger right helix to left helix ratio. Conversely, molecules with an odd number of atoms tend to have almost equal ratios of left to right helices. These are consistent with Aggarwal's finding.

There are several factors that could account for the discrepancies between new findings and previous studies. One possibility is that this study is not examining identical molecules. Additionally, the number of conformers considered in previous studies was significantly smaller, ranging from 27 to 130, compared to the vast datasets generated by *Diamond Energy*, which include up to thousands of outputs for certain molecules. As a result, the total helix ratio calculated by *Diamond Energy* is likely to be more conservative, given the larger dataset.

Figure 2.44 Total helix ratio in three *Diamond Energy* conformer generation results.

2.4 Conclusions

Diamond Energy is an internal coordinate tree-searching algorithm designed for rapid molecular structure analysis based on the diamond lattice framework. Inspired by the geometric properties of diamond lattices, which are characteristic of low-energy conformers, it utilizes simple integer mathematics for the generation of conformers and facilitates swift energy evaluations.

Our tests on acyclic alkanes mark the first demonstration of *Diamond Energy*'s ability to accurately and rapidly identify low-energy structures in comparison with four well-established algorithms: SPMC, MCMM, LMCS, and MCLMCS. Notably, the average runtime of the *Diamond Energy* algorithm for finding a single conformer is significantly less—by more than two magnitudes—than that of the aforementioned algorithms. Moreover, *Diamond Energy* has been found to accurately identify the lowest energy conformers in most test cases, comparable to these methods using both MM3 and OPLS4 calculations. Furthermore, the number of conformers it identifies within the lowest energy ranges of 3 kcal/mol and 5 kcal/mol surpasses that of the four compared methods, demonstrating its capability to efficiently explore the low-energy space of the test molecules.

Overall, *Diamond Energy* proves to be an exceptionally fast and competent new algorithm for conformer search.

Additionally, this method shows promise for the rapid structural analysis of saccharides, which similarly align well with the diamond lattice geometry. These applications will be further elaborated upon in subsequent chapters.

Chapter 3

Further Development of *Diamond Energy*

In preliminary tests involving acyclic alkanes, *Diamond Energy* was found to efficiently and rapidly identify low-energy conformers. These results serve as a proof of concept, suggesting that the resemblance of local minima to diamond lattice structures can aid the internal tree algorithm in developing a robust strategy for conformer searches.

"While *Diamond Energy* was primarily developed for acyclic alkanes, it became evident that chair-shaped six-membered rings also align well with the diamond lattice framework. These rings are crucial in many molecules, especially pyranose sugars. In light of this, the next step involves refining *Diamond Energy* to accommodate these ring structures.

In the next phase of the research, there are plans to further develop the applicability of this algorithm to more complex systems, as shown in Figure 3.1. This is particularly relevant for saccharides, whose skeletal structures also bear similarities to diamond lattices. Conformers of saccharides are crucial to studies of these molecules, as stereoselectivity is a key consideration in saccharide synthesis. Subsequent chapters will detail the methodology and results of applying this algorithm to such complex molecular structures.

As in previous iterations, the program ultimately generates a set of local minima that are ready for subsequent energy minimization procedures. For further clarification, a flowchart delineating the program's modifications is provided on the right (Figure 3.2).

In assembling the saccharides part, modifications were implemented in the program to differentiate between ring and non-ring atoms. All processes in the program stay consistent

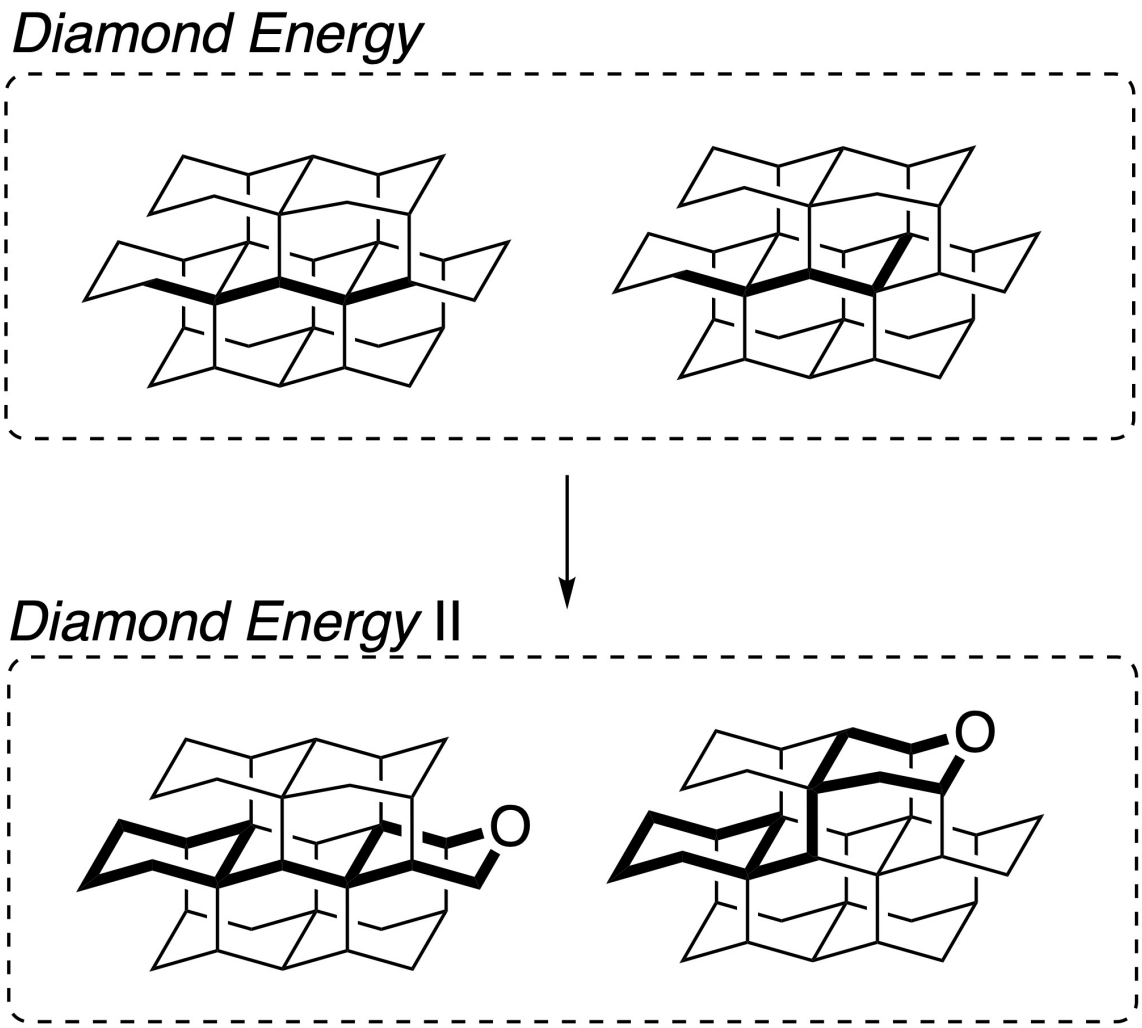


Figure 3.1 Extending *Diamond Energy* to Complex Saccharide Systems: Structural Affinities with Diamond Lattices.

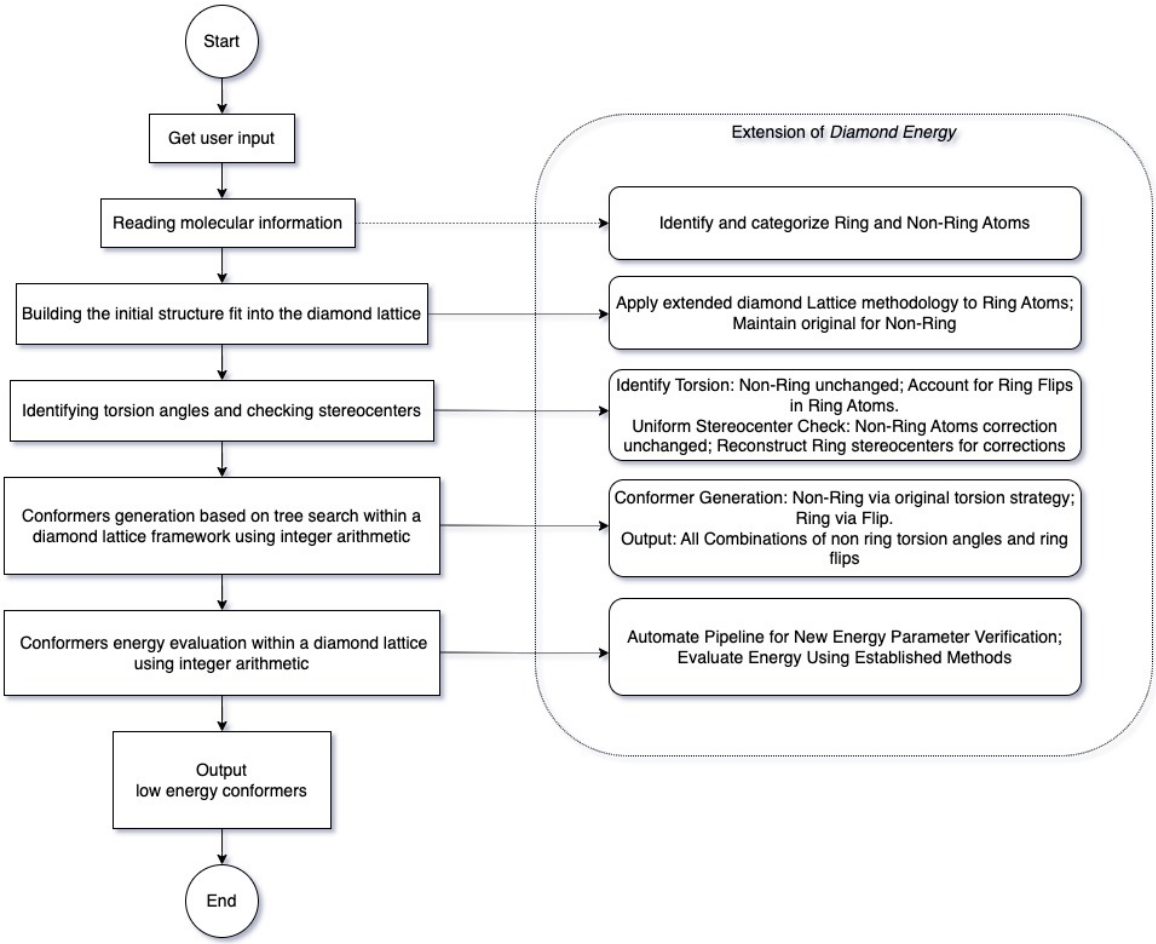


Figure 3.2 *Diamond Energy*: Upgrades and New Features in Version II.

when applied to non-ring atoms. However, there are some additional strategies that are specifically used for ring atoms.

In the initial structure-building phase, non-ring components are generated using the established alkane strategy, while ring components are constructed based on a modified version of this strategy, specifically tailored to close the ring structure. For ring structures, initial ring twist values which are used to close the ring structure are precalculated and stored in lists, along with corresponding values for ring-flipped configurations. For ring conformations, the current version is limited to two specific forms: the chair shape and its ring-flipped counterpart due to the diamond lattice framework.

Subsequent steps involve the evaluation of stereocenters. Non-ring atoms are evaluated for stereocenters using the original *Diamond Energy* strategy. For ring structures, if the stereocenter is identified incorrectly, the ring is reconstructed using a precalculated twist value list.

The final phase consists of the geometry generation. For non-ring parts, a tree search is conducted with torsion angle as the variable. For ring parts, the tree search takes ring flip as a variable.

Additionally, because systems similar to saccharides often contain atoms like oxygen, an automatic pipeline has also been developed to evaluate the potential energy terms associated with the introduction of new atoms.

Essentially, *Diamond Energy* II implements changes to enable the program to handle cyclohexane and the introduction of oxygen atoms. The subsequent sections will provide detailed illustrations of these changes.

3.1 Incorporation of Cyclohexyl Groups into the Model

In the process of assembling saccharides, the existing framework of the *Diamond Energy* program remains largely unchanged, with the exception of specialized treatment for ring atoms.

In the program, molecular structures are constructed from one end to the other, based on the connectivity information in the InChI notation (Figure 3.3). To streamline ring

assembly, the program continues in the same direction when it encounters ring atoms during the construction of an alkane chain. For example, with methylcyclopentane, the program reads its InChI, 'InChI=1S/C7H14/c1-7-5-3-2-4-6-7/h7H,2-6H2,1H3', and follows the InChI sequence '1-7-5-3-2-4-6' to build the structure. Since atom 7 appears more than once, the program ignores the duplicate. It also records that atom 1 is a non-ring atom, while the others are part of a ring. The initial structure is built in a beta configuration, with ring atoms positioned to form a six-member ring within a diamond lattice framework. The correct configuration of the ring is not confirmed at this stage, and stereocenters are adjusted later if necessary. This approach, which has been effective for non-ring atoms, is applied to the cyclohexyl substructure: build quickly first, then revise the stereocenters as needed. Methods for closing the ring are precalculated and stored for retrieval, allowing for the initial construction and any necessary subsequent corrections to be completed quickly.

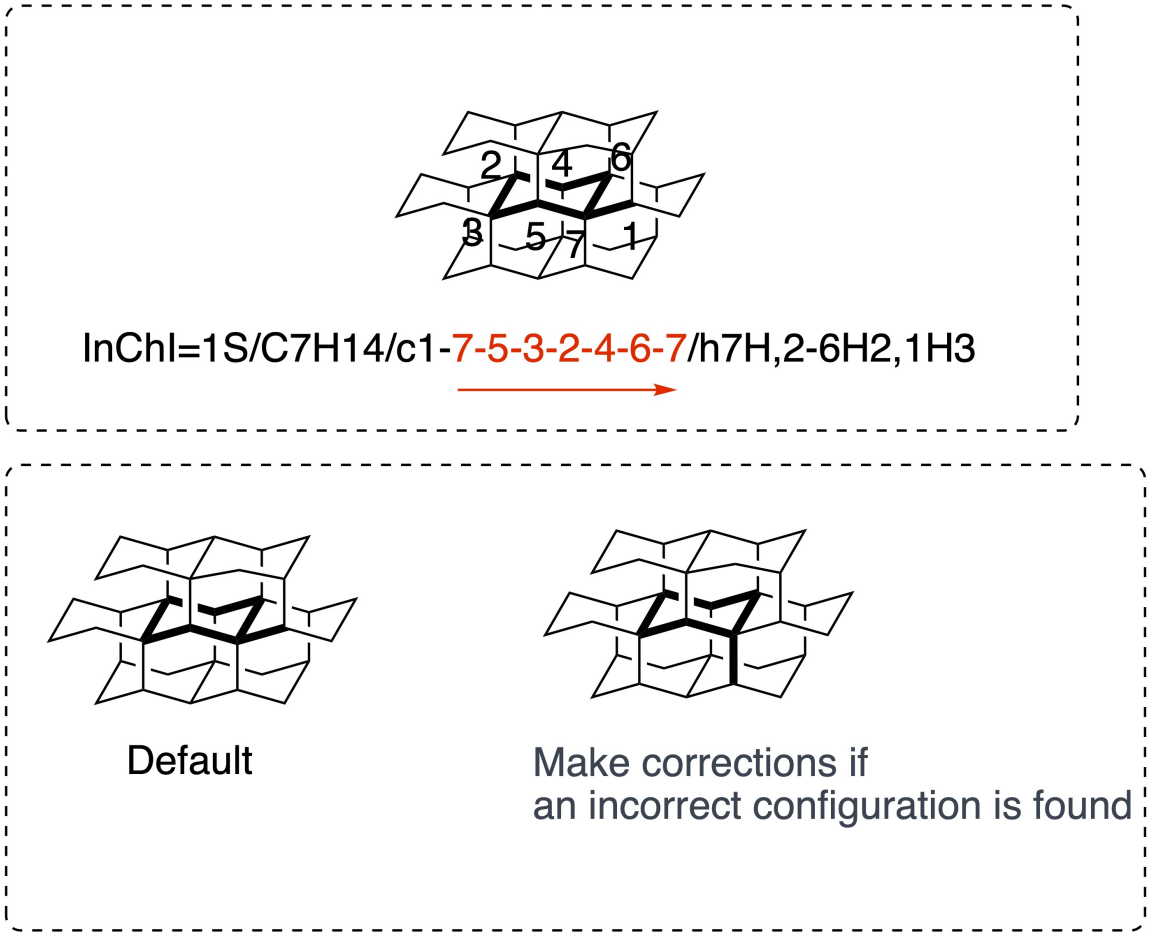


Figure 3.3 An example of how *Diamond Energy* II builds the structure of methylcyclopentane.

In addition to constructing ring structures, the assembly of the remaining atoms will proceed using the original strategy. Since there are no changes to that part, it will not be detailed further here; for reference, please see the previous section.

As the program assembles the initial structure, it also identifies the molecule's variables, which are essential for generating subsequent conformers. The variables used for generating geometry include rotatable bonds for non-ring atoms and the entire ring for ring structures. Thus, during the conformer search phase, the rotatable bonds of non-ring atoms are adjusted to generate new positions, whereas ring structures undergo a ring flip. To clarify this process, simple examples are illustrated in Figure 3.4.

After the initial structures have been constructed, the next step is to examine the stereocenters. In the original algorithm designed for acyclic alkanes, if a stereocenter is built incorrectly, it can be corrected, without impacting the rest of the structure, as the construction is verified sequentially from one end to the other. However, in ring systems that form closed loops, an incorrect change to the structure involving a ring could potentially break it. Although the code for ring systems still follows the InChI sequence and examines atoms one by one, correcting an incorrect atom in a ring affects the whole structure. Therefore, it's crucial to differentiate between ring and non-ring atoms when correcting errors at stereocenters. The code for non-ring atoms part does not change and will not be detailed here again.

This is how the program checks and corrects stereocenters involving rings. For atoms within a ring, the type of correction needed depends on the atom's position. If the first atom of the ring is incorrect, the entire structure needs to be rebuilt in a new configuration, like changing from beta to alpha, using a pre-calculated list. If the error is at a different atom, only the affected branch at the stereocenter is adjusted to maintain the ring's integrity, without having to rebuild the whole ring. The processes are illustrated in Figure 3.5 using two simple examples.

After assembling the molecular structures, the program proceeds to generate conformers. It uses a tree-searching algorithm to examine all possible combinations of variables, as it did before. The main change is the addition of variables for ring reconstruction. The program can rotate the entire ring to preserve the chair conformation of a six-membered ring or transform it into a different chair conformation, adhering to the diamond lattice structure. These ring conformations are readily available from a pre-stored list. While the energy evaluation process is mostly the same, special consideration is now needed for overlapping atoms and

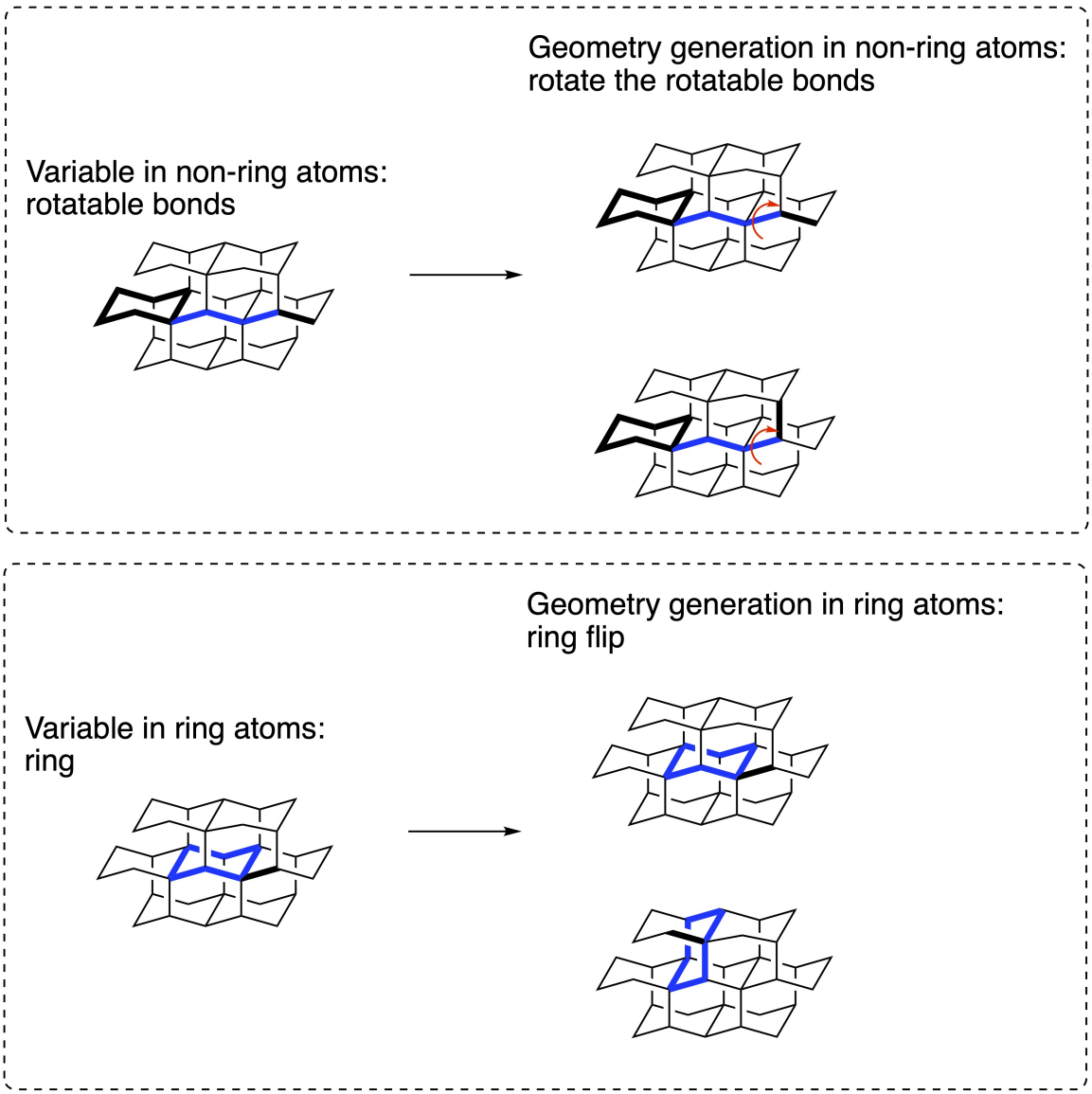


Figure 3.4 Illustration of *Diamond Energy* II Variables and Their Adaptations to Generate New Conformers.

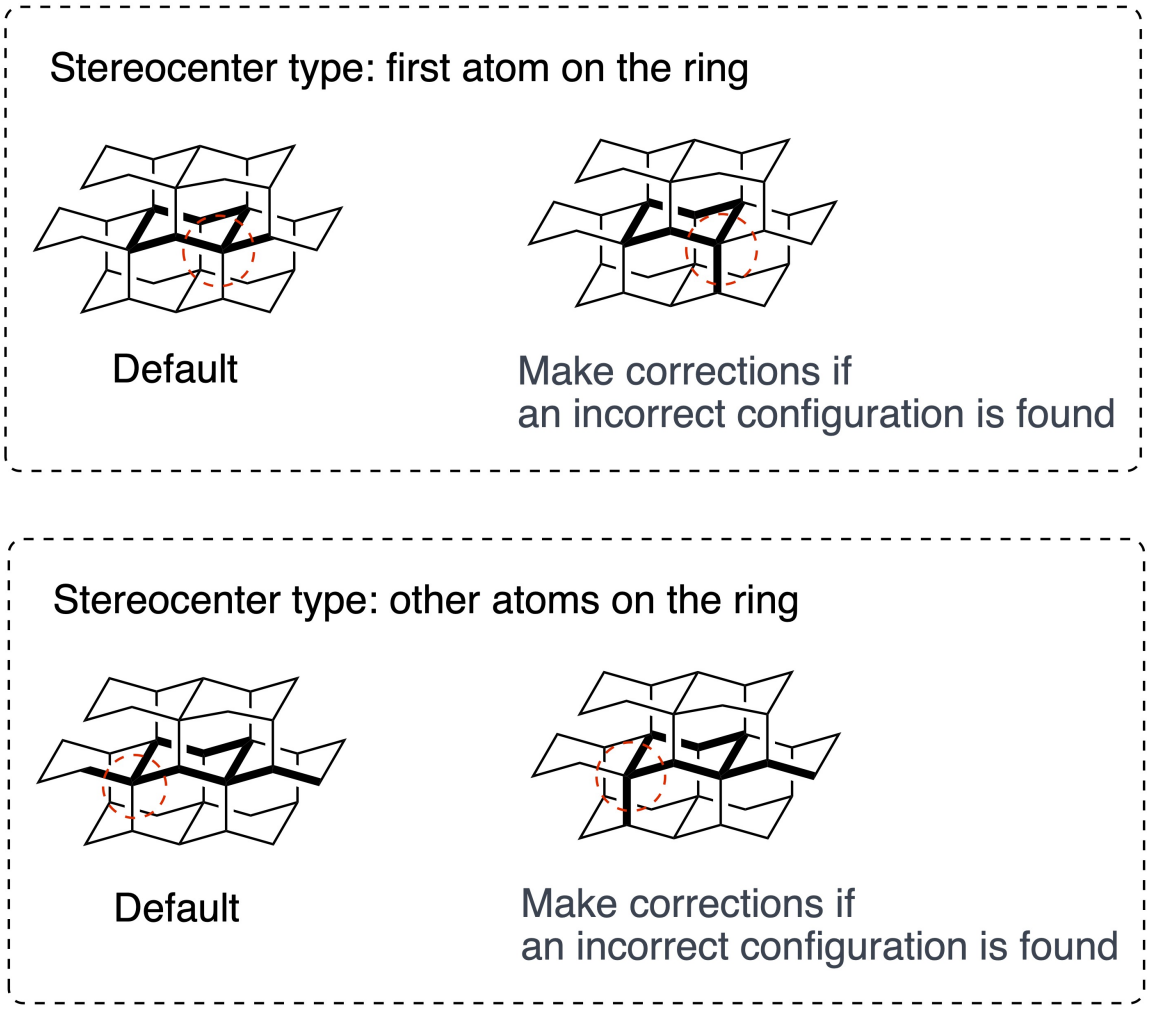


Figure 3.5 Illustration of *Diamond Energy* II Checking and Correcting Stereocenters.

proximity issues that arise from ring-based non-bonding interactions. To better illustrate this process, Figure 3.6 is shown below to present how it works.

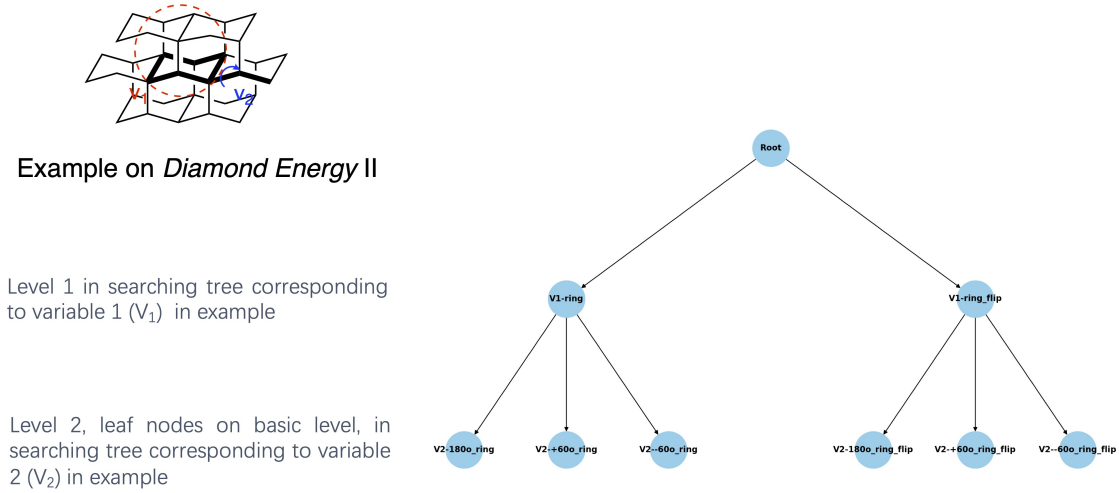


Figure 3.6 *Diamond Energy* II searching tree example.

3.2 Incorporating Oxygen Atoms into the Model

The script begins by reading the molecular information from its input InChI and then processes and stores essential details such as atomic numbers, connectivity, stereocenters, etc. Recognizing oxygen is readily accomplished by utilizing the related functions within the RDKit package.

The main change after introducing oxygen is the method of evaluating molecular energy. The script calculates the total energy of a conformer by iterating through the molecular structure and adding up the interaction energies between each pair of atoms. With the addition of new types of atoms like oxygen, different interaction patterns may emerge.

To estimate the range of new possibility parameters, the script initially treats oxygen atoms as carbon atoms, without distinction. This approach tests if the existing energy terms for carbon are still applicable to interactions involving oxygen. If they are, no new energy terms are necessary. If not, the values must be manually adjusted to find an energy range that works.

3.2.0.1 Establishing Approximate Ranges for New Parameters

We started with the assumption that the new oxygen atom behaves similarly to carbon in terms of its energy patterns when not bonded to other atoms. Therefore, we applied the energy values that were determined for carbon to these non-bonded interactions involving oxygen (Figure 3.7). The aim is to evaluate whether these original values remain effective in guiding the diamond lattice energy model toward identifying low-energy conformers. For this evaluation, MCLMCS, the default conformer search algorithm in MacroModel, will be used as a benchmark.

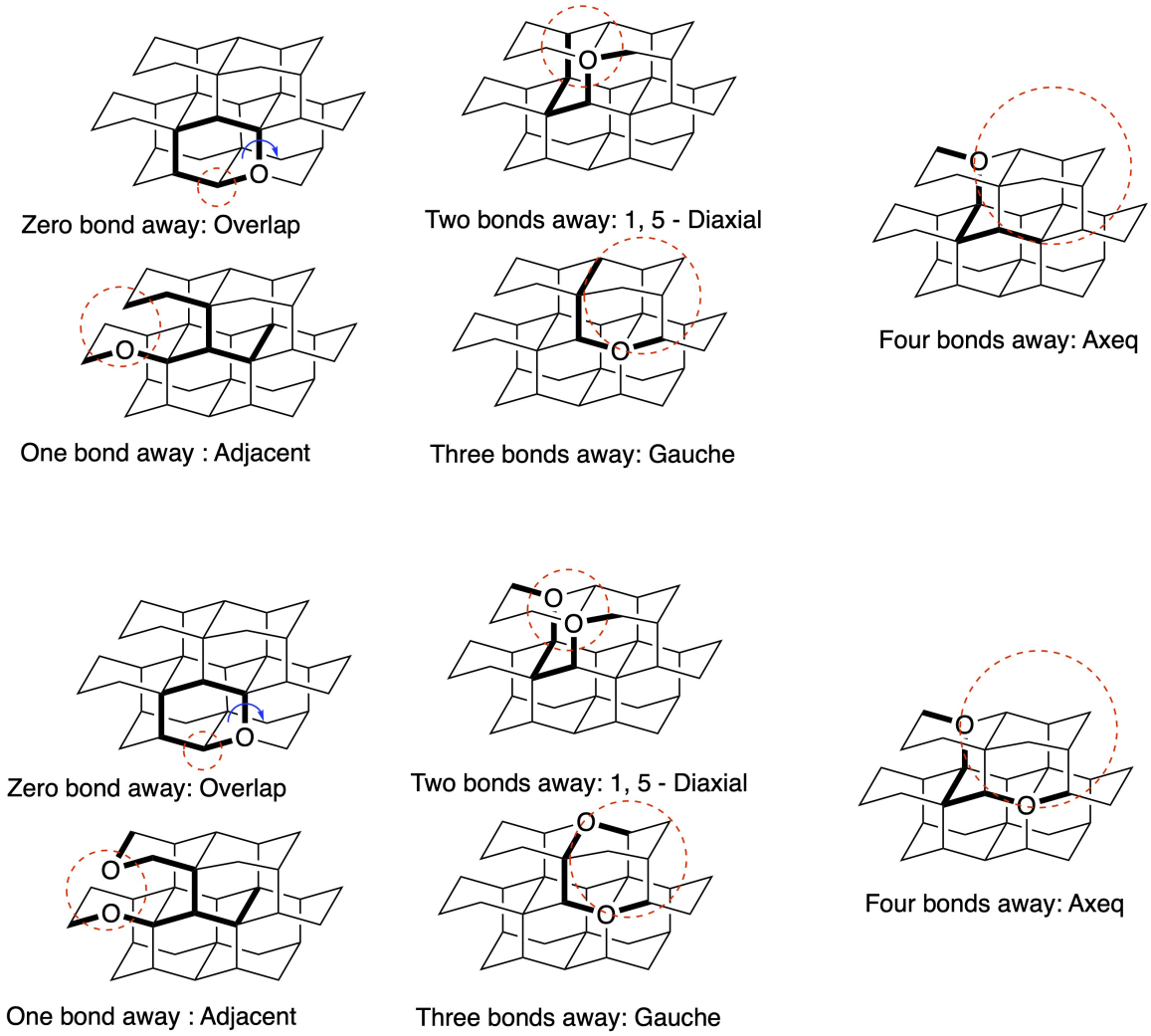


Figure 3.7 Assumed Non-Bonded Interactions for Newly Introduced Oxygen Atoms.

To evaluate the parameters for carbon and oxygen interactions, a series of simple molecules are used to examine how the energy terms originally defined for carbon apply to interactions

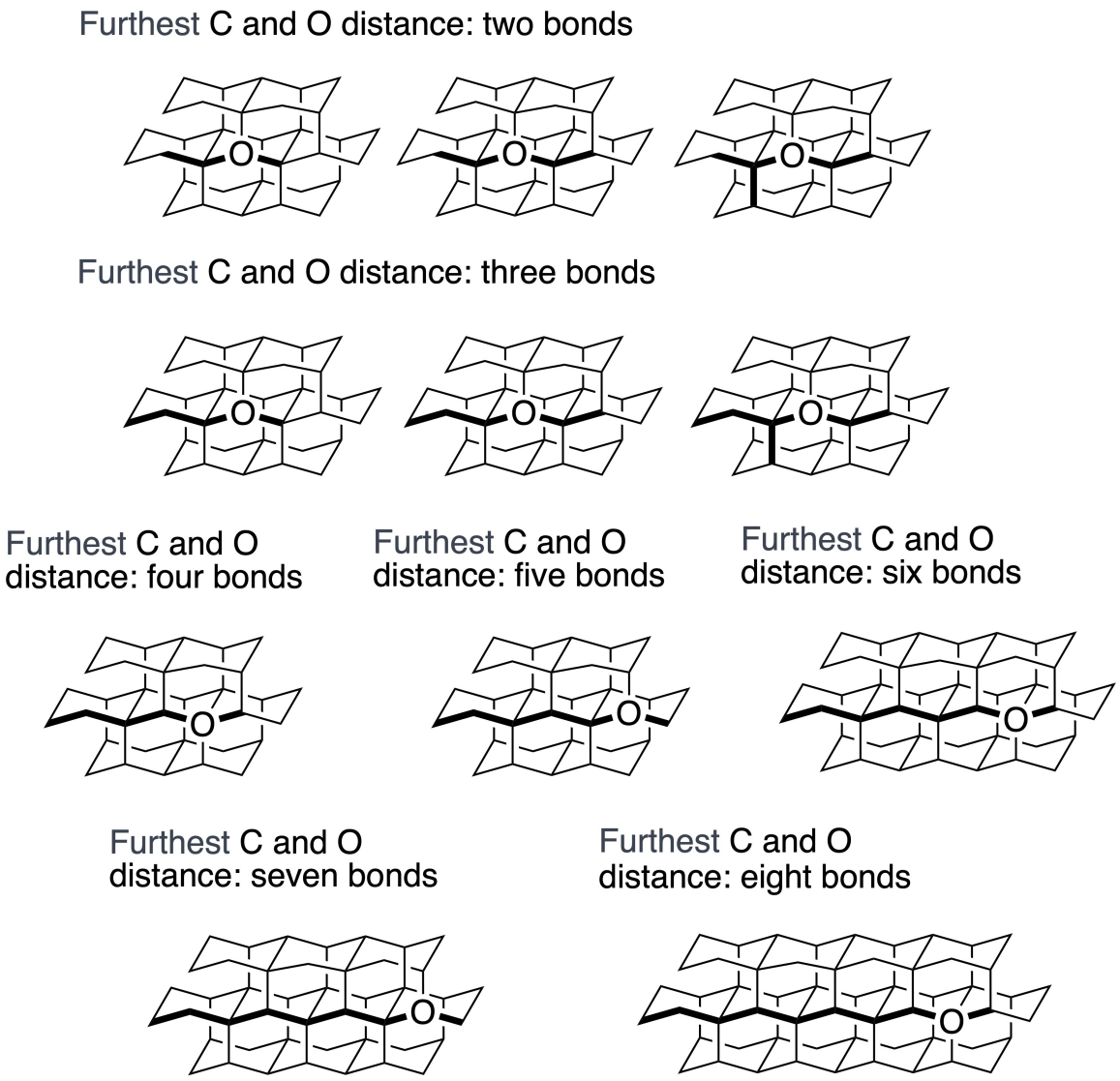


Figure 3.8 Molecules Selected for Testing Approximate Parameter Ranges.

involving both oxygen and carbon. These molecules primarily consider the farthest distances between carbon and oxygen atoms, ranging from two to eight bonds apart as illustrated in Figure 3.8. Distances greater than eight bonds are not considered necessary to examine, as they are well-integrated into the diamond lattice structure and are covered by existing interaction considerations.

When the test was conducted on the molecule COCCC, the results obtained using the initial parameters in the *Diamond Energy* model deviated from those generated by MacroModel. This discrepancy arises due to the presence of a unique interaction involving oxygen. Specifically, the gauche conformation, which is generally unfavorable in C-C interactions, becomes favorable in C-O interactions due to the formation of a hydrogen bond (see Figure 3.9).

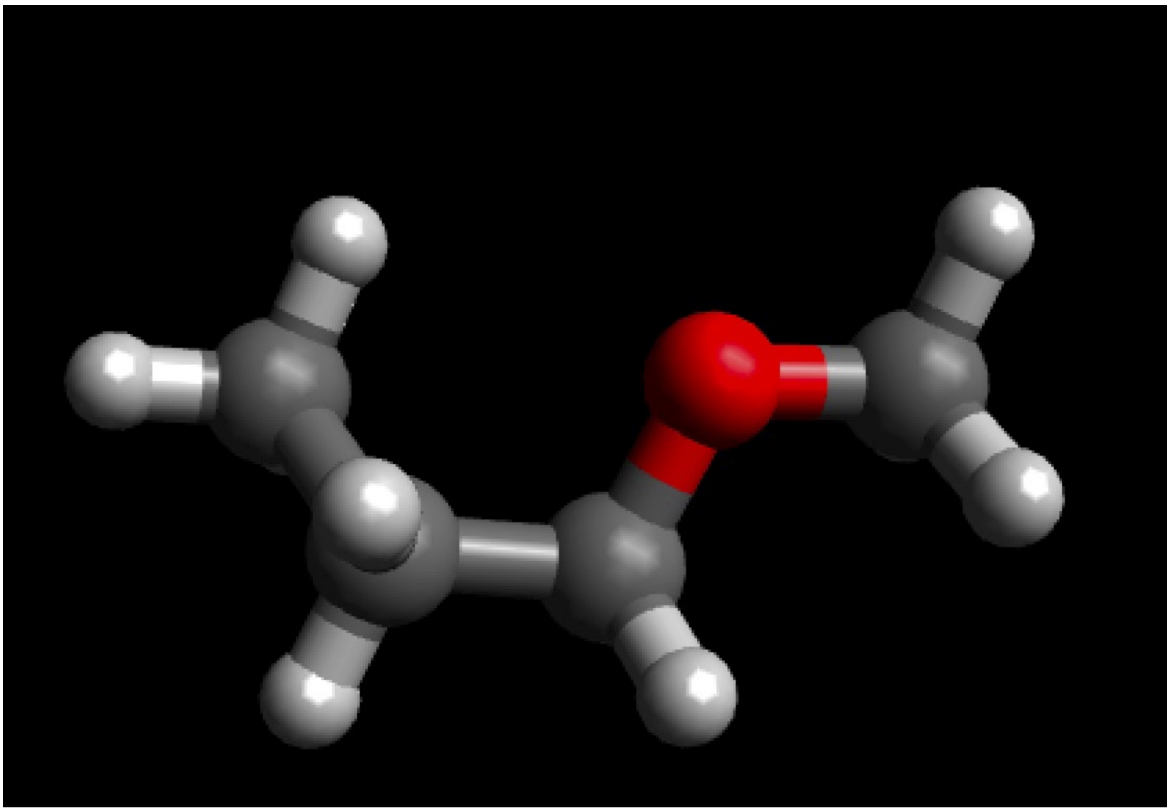


Figure 3.9 Identification of a Different Interaction Motif in C-O Example.

In adjusting the C-O gauche term value, a decrement of one was applied, starting from an initial value of 2. Once the value reached -2, the outcomes from the *Diamond Energy* model and MacroModel were found to align. Consequently, a rough range of -1 to -4 is considered suitable for the C-O gauche term.

Following this methodology, other interaction term values were also modified. As the complexity of the test set molecules increased, no additional motifs were identified. The provisional range for C-O interactions is summarized as follows: 1,4-gauche (-4 to -1), 1,5-diaxial (-7 to -2), and axial-equatorial (-4 to -1). The thresholds for overlap and adjacent interactions remain at the original value of 200.

The presence of potential interactions between oxygen atoms was also investigated in the same way. To examine this, a set of molecules featuring a variety of oxygen-oxygen bond distances, ranging from 2 to 8 bond distances, was generated (Figure 3.10). Tests were initially run on these molecules using parameters for carbon-carbon interactions as a starting point. Through this iterative process, a preliminary range for oxygen-oxygen interactions has been established: 1,4-gauche (-8 to -4), 1,5-diaxial (5 to 15), and axial-equatorial (-4 to -1). The thresholds for overlap and adjacent interactions remain set at the original value of 200, approaching 200 kJ/mol, which indicates a very high energy level. Conformers exhibiting this high energy level are considered unfavorable.

Beyond the previously mentioned ether oxygen, alcohol-type oxygen may also lead to new interaction motifs. Therefore, the same method used to establish a rough range for new energy terms involving ether oxygen should be applied to alcohol oxygen to determine its interactions. It is hypothesized that alcohol oxygen exists within the same interactions as the carbon-only energy term. Utilizing the same energy term for molecules that involve alcohol oxygen, it can then investigate whether these energies assist *Diamond Energy* II in identifying low-energy conformers, as the benchmark method MCLMCS does. If it does not, then it will need to modify the energy term involving the test molecule manually until a rough range is found, where the energy term can aid in identifying low-energy conformers, as the benchmark method does.

Similarly, molecules with varying distances between carbon and alcohol oxygen pairs, alcohol oxygen itself pairs, and ether oxygen and alcohol oxygen pairs are drawn and used for this test, as shown in Figure 3.11. After conducting the above test, the rough range for the new energy terms is confirmed as follows: axeq_C_OH (-2 to -1), gauche_C_OH (-3 to -2), 1,5diaxial_C_OH (1 to 3), axeq_O_OH (-2 to -1), gauche_O_OH (-4 to -2), 1,5diaxial_O_OH (-15 to -10), axeq_OH_OH (-9 to -7), gauche_OH_OH (-5 to -4), and 1,5diaxial_OH_OH (-20 to -15). Additionally, four new correction motifs have been identified: adjacent_O_OH (-27 to -25), adjacent_OH_OH (-30 to -28), 1,3_connect_O_OH (-20 to -18), and 1,3_connect_OH_OH (-22 to -20).

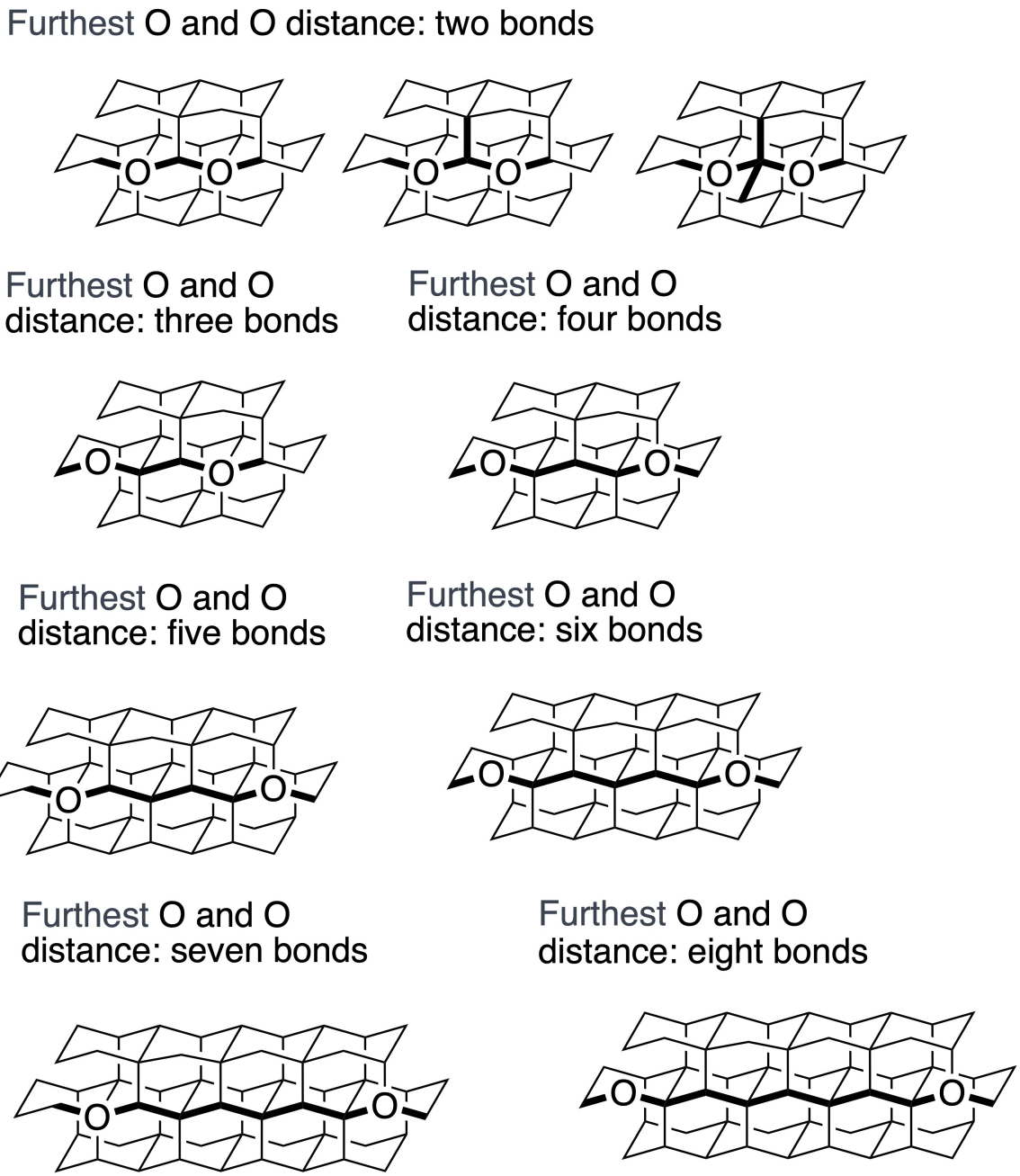


Figure 3.10 Test Molecule Set of O-O Interaction Parameters.

CCO	OCCO	COCO
CCCO	OCCCO	COCCO
CCCCO	OCCCCO	COCCCO
CCCCCO	OCCCCCO	COCCCCO
CCCCCCO	OCCCCCCO	COCCCCCO
CCCCCCCO	OCCCCCCCO	COCCCCCCO
CCCCCCCCO	OCCCCCCCCO	COCCCCCCCO
	OCCCCCCCCCO	COCCCCCCCCO
		COCCCCCCCCCO

Figure 3.11 SMILES Representations of Molecules Used to Test the Broad Range of New Energy Terms Introduced with Alcohol Oxygen.

With this range for each new parameter determined, the next step is to establish the precise value for each term and to find the best combination of these terms. Details of this process will be outlined in the following sections.

3.2.0.2 Employing Bayesian Optimization to Determine Potential Parameter Combinations Within the Defined Range

To identify an optimal set of parameters within the previously established range, a Bayesian optimization algorithm was used to aid the process. The process is outlined in Figure 3.12.

To expedite the optimization, the script compares the energy values of 25 lowest-energy conformers identified by both methods. The primary objective of this hyperparameter optimization is to verify that the chosen parameter set is capable of accurately identifying the global energy minimum, consistent with the results from the MacroModel default method, MCLMCS. Given that MCLMCS is presumed to explore the low-energy space of the molecular test set effectively, the evaluation criterion is based on the R-squared value between the two methods' lowest 25 energy values outcomes.

The process of identifying optimal combinations of energy parameters can be aptly framed as a hyperparameter optimization problem. Traditional methods for hyperparameter tuning include Grid Search and Random Search. Grid Search involves training models on all possible combinations of hyperparameters and then selecting the most effective set based on predetermined performance criteria. While straightforward, this method is computationally

expensive. In contrast, Random Search samples hyperparameters from a uniform distribution, offering greater computational efficiency, face the potential cost of less precise optimization.

Emerging methods have further refined the hyperparameter optimization process. Gradient-based Optimization is another method, ideal for neural networks, which uses gradient descent techniques to update hyperparameters. Evolutionary Algorithms, like Genetic Algorithms, also offer a different paradigm for optimization by mimicking natural selection processes such as inheritance and mutation. Bayesian Optimization employs a probabilistic model-based approach to strike a balance between exploring new hyperparameter spaces and exploiting known optimal spaces.

Given the complexities of the energy parameter optimization problem, where the objective function can be considered a 'black box,' Bayesian Optimization is regarded as the most suitable approach for this study. Unlike other methods, Bayesian Optimization is well-equipped to handle such black-box optimization scenarios effectively.

The optimization process, as depicted in Figure 3.12, involves a parallel conformation search for a given test set of molecules. The MacroModel's default conformation search algorithm, MCLMCS, followed by the default PRCG minimization, serves as a benchmark. In contrast, another process utilizes *Diamond Energy* II with an energy term parameter set selected from the range defined in the previous section, followed by minimization using the MacroModel's default PRCG method. Upon completion of each round, the lowest 25 energy values from both pipelines are compared to calculate their R^2, indicating the closeness of the results obtained using the chosen parameter set in *Diamond Energy* to the benchmark method's outcomes. If the R^2 is not deemed sufficiently converged, a new parameter set is derived using the Bayesian Optimization package for further testing. This iterative process continues until the R^2 is considered to be converged, at which point the optimization is halted.

The test was first applied to confirm the six energy parameters involving ether oxygen: 1,4-gauche_C_O (-4 to -1), 1,5-diaxial_C_O (-7 to -2), axial-equatorial_C_O (-4 to -1), 1,4-gauche_O_O (-8 to -4), 1,5-diaxial_O_O (5 to 15), and axialequatorial_O_O (-4 to -1). A test set of molecules are selected for the optimization process (Table 3.1). These molecules were manually selected to cover a wide range of non-bonded interaction motifs, ensuring an even distribution across the test set. Frequency and similarity analyses of this dataset are presented in Figures 3.13 and Figure 3.14. The analyses show that the dataset is diverse and the motifs are well-represented.

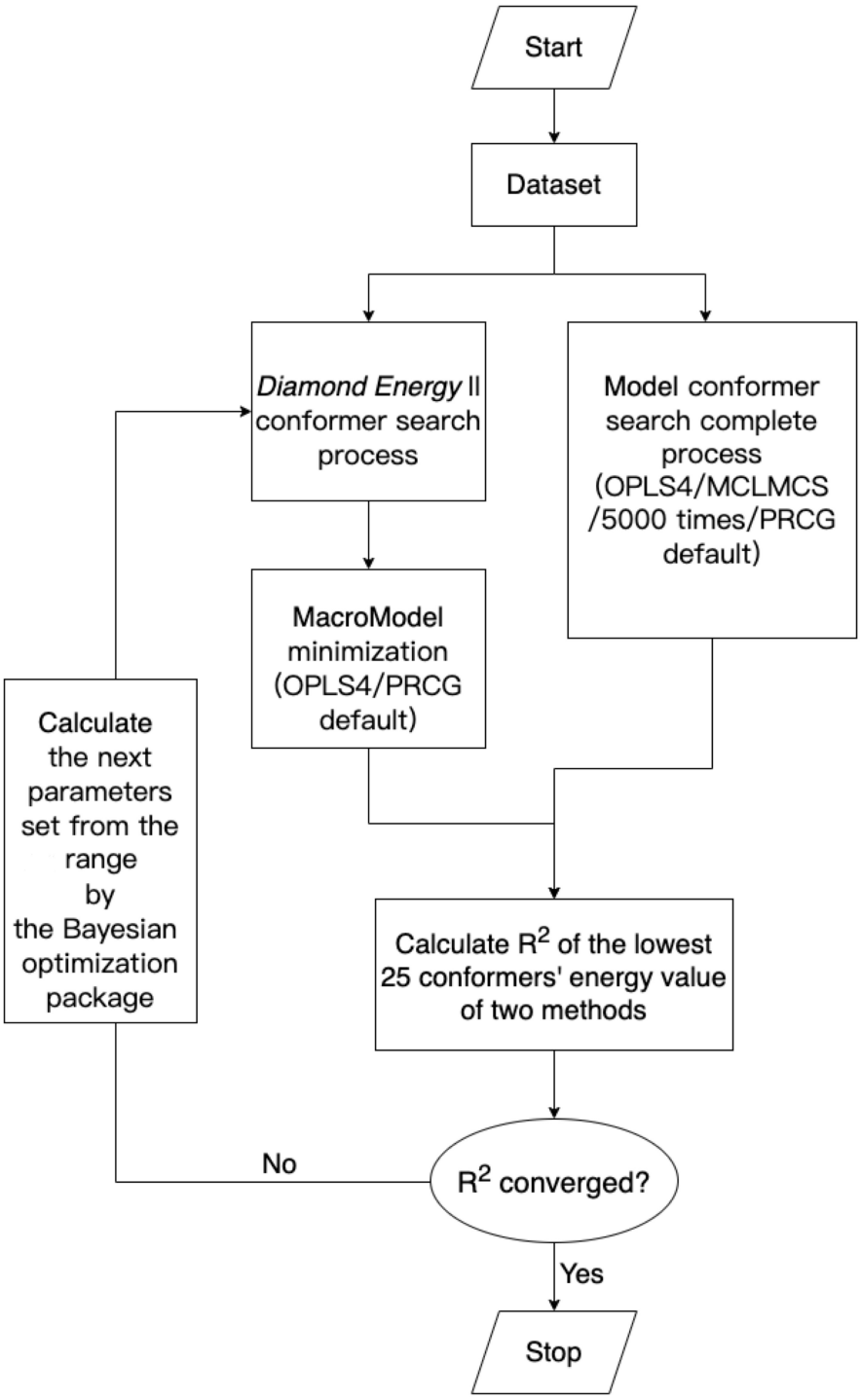


Figure 3.12 Flowchart Depicting the Process of Parameter Set Optimization Based on Bayesian Optimization.

Table 3.1 Hyperparameters Verifying Dataset Used for Ether Oxygen Related Terms.

CCOC
COCCC
COCCCC
COCCOC
COCCC(COC)OC
COCOC
COCCCOC
COCCOCC
COCOCOC
COCOCCOC
COC(COC)CC
COCC(CC)C(COC)CC
COCC(C)C
COC(C)OC
CCOCCCC
CCOCCCOCC
CCOCCC(OCC)COC
COCOCOCOCOCC
CCOCC(COCC)C
CCOCC(COCCOC)CC
COC(CC(C)C)COC
CC(C(C)OCC(C)COC)C
COCC(C)COC(C)(C)C

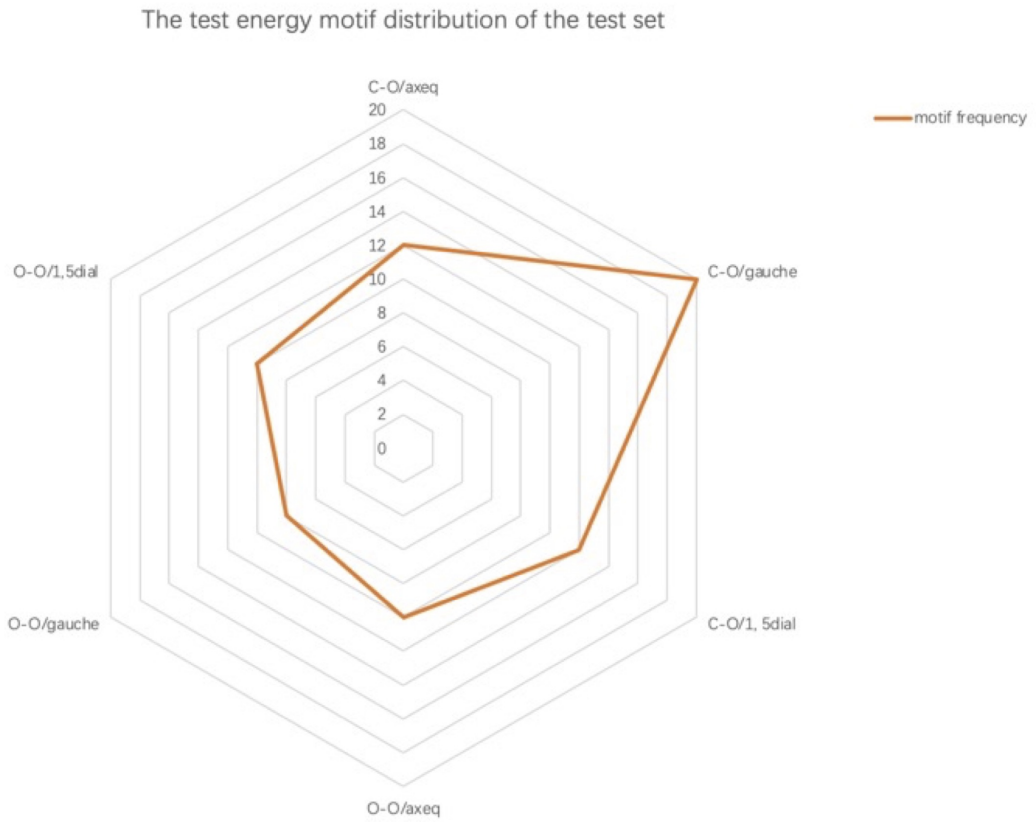


Figure 3.13 Motif Frequency Distribution Analysis of the Test Set for Ether Oxygen Related Parameters.

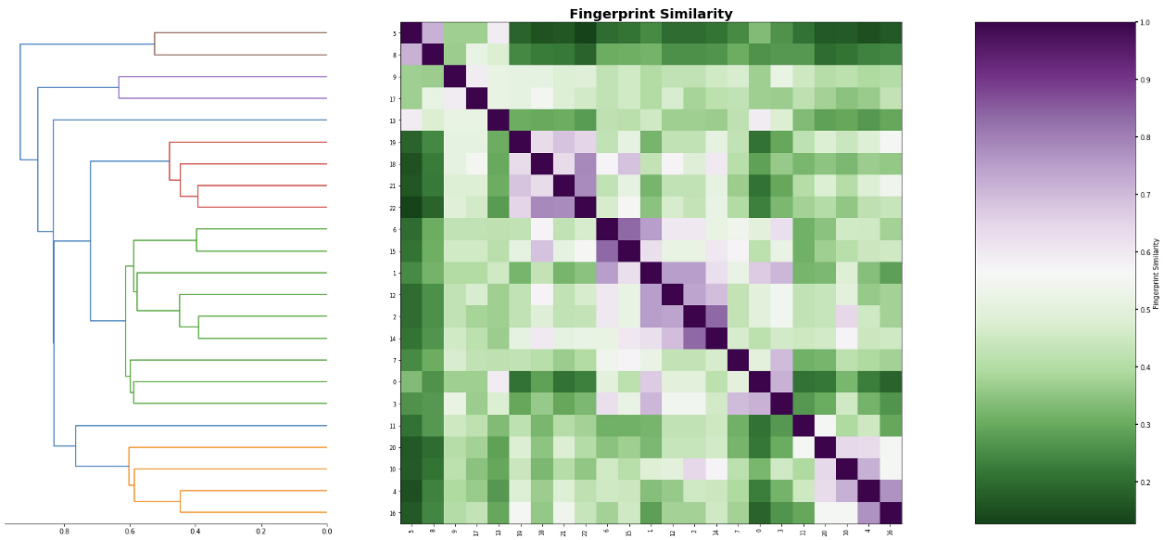


Figure 3.14 Hyperparameters Verifying Dataset Similarity Analysis for Ether Oxygen Related Parameters.

The most promising results to date have been obtained R^2 values consistently ranging between 0.92 and 0.94. This suggests that the optimization may be nearing convergence. To determine whether the parameter set with the best R^2 should serve as the final solution, the following verification process is conducted.

The primary objective of optimizing the parameter set is to ensure its ability to effectively locate the global energy minimum, akin to the default method, MCLMCS in MacroModel. Given that MCLMCS is assumed to thoroughly explore the low-energy conformational space of the test set of molecules, the R^2 values of the 25 lowest-energy conformers identified by both methods serve as the evaluation criterion.

If the current best-performing parameter set enables *Diamond Energy* II to match or exceed MacroModel's performance across all test motifs and throughout the entire search space, then this parameter set can be accepted. To validate this, the leading parameter set is applied to a set of molecules originally used for testing all possible energy motifs. The comparative effectiveness of *Diamond Energy* II and MCLMCS in exploring the overall conformational space is then assessed via a performance score.

To evaluate the exploration of the conformational search space, a comparison score is introduced for the test sets analyzed by both *Diamond Energy* II and MCLMCS. Assuming all conformations adhere to a Boltzmann distribution, this approach assesses how closely the combined results of the two methods approximate the theoretical baseline — an aggregation of energy values derived from both methods.

The unique energy values obtained from the Diamond script are labeled 'Result Diamond Unique', while those exclusively from the Maestro script (MacroModel) are labeled 'Result Maestro Unique'. The combination representing the 'Best Approximation' is the sum of unique values from both methods. This 'Best Approximation' will then be used for further calculations (Figure 3.15).

According to the Boltzmann distribution, the probability of a certain energy value found should be:

$$P_i = \frac{e^{-\frac{E_i}{kT}}}{\sum_j e^{-\frac{E_j}{kT}}}$$

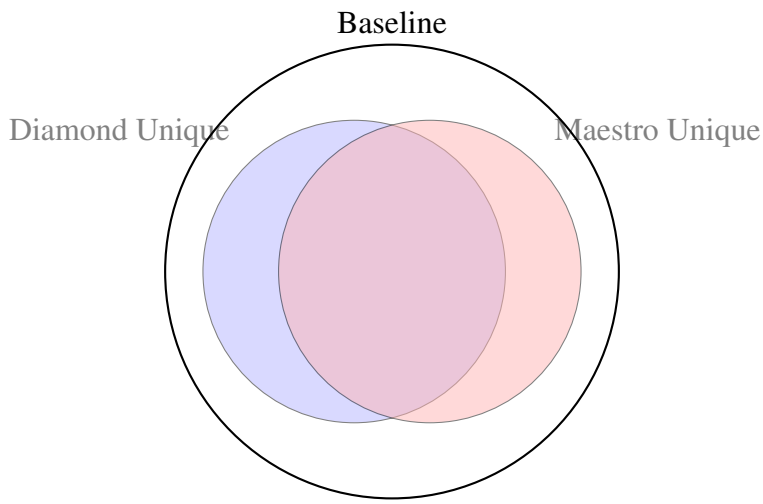


Figure 3.15 A diagram showing the relationship between Diamond Unique (purple), Maestro Unique (orange), and the Baseline. The evaluation score used in this study will utilize the sum of Diamond Unique and Maestro Unique as the best approximation for calculation.

So, a performance score P to represent how well the space has been searched is introduced, utilizing this probability as a weight term and adding up all this weight to show how many low energy spaces have been fully explored:

$$\text{Performance score of a method } P = \sum_{i=0}^{M=\text{total_num}} P_i$$

For simplicity, each energy's P_i calculation, related energy value is used instead of absolute energy, as absolute energy values vary in different fields. As the ratio of probabilities of two states is known as the Boltzmann factor and characteristically only depends on the states' energy difference:

If it takes the lowest energy as E_{ref}, for any given energy, its Q_i could be calculated:

$$Q_i = e^{-\frac{\Delta E}{RT}}$$

And its P_i will be:

$$P_i = \frac{Ce^{-\frac{\Delta E}{RT}}}{C\sum_{i=0}^{\text{num=total}} e^{-\frac{\Delta E_i}{RT}}} = \frac{e^{-\frac{\Delta E}{RT}}}{\sum_{i=0}^{\text{num=total}} e^{-\frac{\Delta E_i}{RT}}}$$

In this way, the comparison of the two methods will rely on comparing their performance scores, which sum up the probability of each energy value found under the Boltzmann distribution as a weight term:

$$\text{The performance score of a method } P = \sum_{i=0}^{M=\text{total_num}} P_i$$

(P_i is the probability value of an energy value found)

The parameter set value associated with the stable R^2 obtained by the automatic pipeline was then used to calculate the Boltzmann factor-based performance score for verification. Both the lowest energy values found by the two methods and their performance scores are very close for the test cases, as shown in Table 3.1 and Table 3.2. While there are cases where the values differ, the gap is minimal. Therefore, the energy terms obtained through Bayesian optimization so far are considered acceptable.

The comparison results of the two methods, using the current best parameter set found by Bayesian optimization by R^2 criteria, are listed in Table 3.2. The third and fourth columns present the lowest energy values found by *Diamond Energy* II and MCLMCS, respectively. It is evident that in most cases, both methods yield identical values. However, there are several instances where differences arise between the two methods, but the gap in values is minimal. The subsequent columns display their corresponding performance scores, calculated based on the Boltzmann factor. A higher score signifies better performance and is also emphasized in bold. As can be seen, in the majority of test cases, the two methods share the same performance score. Yet, in some molecules, there are slight differences in performance scores between the methods, though these differences are marginal. From these results, it can infer that the parameters for ether oxygen-related terms found so far are considered acceptable.

Similarly, Bayesian optimization was conducted for the energy terms related to alcohol oxygen. Molecules with various motifs involving the energy term to be tested were selected, as listed below (Table 3.3). The R^2 values remain consistently stable between 0.95 and 0.97, with the highest R^2 being 0.974. The parameter set associated with these results was then used to calculate the Boltzmann factor-based performance score for verification, with the results shown in Table 3.4. Similarly, both the lowest energy values found by the two methods and their performance scores are very close for the test cases. While there are cases where the values differ, the gap is minimal. Therefore, the energy terms related to alcohol oxygen obtained through Bayesian optimization are considered acceptable.

Table 3.2 Lowest Energy and Boltzmann Factor based Performance Score Comparison between *Diamond Energy* II and MCLMCS for Ether Oxygen Related Energy Terms (OPLS4 kcal/mol).

Index	SMILES	Global_min_Diamond	Global_min_MCLMCS	P_Diamond	P_MCLMCS
0	CCOC	−16.33	−16.33	1.00	1.00
1	CCOCC	−52.35	−52.35	0.49	0.51
2	CC(C)OCC	−104.05	−104.05	0.50	0.50
3	CCCOC	11.97	11.97	1.00	1.00
4	CCCOCC	−24.32	−24.34	0.47	0.53
5	CCCOCCC	3.55	3.55	1.00	1.00
6	CCC(C)OC	−34.11	−34.11	1.00	1.00
7	CCCCOC	8.62	8.62	1.00	1.00
8	CCCCCOC	11.29	11.29	1.00	1.00
9	CCCCCCOC	14.00	14.00	0.50	0.50
10	CCCCCCCOC	16.63	16.63	0.51	0.49
11	CCCCCCCCOC	19.20	19.20	0.50	0.50
12	COOC	−20.67	−20.67	1.00	1.00
13	COCOC	−0.86	−0.86	1.00	1.00
14	COC(C)OC	−56.43	−56.43	1.00	1.00
15	COC(C)(C)OC	−175.67	−175.67	0.50	0.50
16	CCOCOCC	−48.95	−48.92	0.54	0.46
17	COCCOC	69.51	69.51	1.00	1.00
18	COCCCOC	−44.16	−44.17	0.48	0.52
19	COCCCCOC	9.31	9.31	1.00	1.00
20	COCCCCCOC	7.05	7.05	1.00	1.00
21	COCCCCCCOC	10.08	10.08	1.00	1.00
22	COCOCOC	−6.97	−6.97	1.00	1.00
23	COCOCOCOC	−17.04	−17.04	1.00	1.00
24	COCOCCCOC	−55.24	−55.25	0.49	0.51
25	CC(C)(OC)C	−141.11	−141.10	0.52	0.48
26	CC(C)(OC(C)C)C	−223.67	−223.67	0.66	0.34
27	CC(OC(C)C)C	−154.40	−154.38	0.53	0.47

Table 3.3 Hyperparameters Verifying Dataset Used for Alcohol Oxygen-Related Terms.

SMILES
OCCCC
OCCCCC
CCC(C)O
CC(C)C(C)O
OCCOC
OCCO
OCCCO
OCC(O)CO
CC(O)CO
OCCCOC
OCCCCCCOC
OCCCCCCO
OCCCCCCCO

Table 3.4 Lowest Energy and Boltzmann Factor based Performance Score Comparison between *Diamond Energy* II and MCLMCS for Alcohol Oxygen Related Energy Terms (OPLS4 kcal/mol), and the corresponding molecules shown in Table 3.3.

Order	Global_min_Diamond	Global_min_MCLMCS	P_Diamond	P_MCLMCS
0	2.07	2.08	0.47	0.53
1	2.62	2.62	0.42	0.58
2	−14.36	−14.36	0.58	0.42
3	−11.56	−11.56	0.65	0.35
4	15.24	15.24	0.50	0.50
5	16.08	16.08	0.67	0.33
6	−7.21	−7.21	0.67	0.33
7	21.37	21.37	0.50	0.50
8	1.98	1.98	0.50	0.50
9	−10.09	−10.09	0.50	0.50
10	0.86	0.86	0.50	0.50
11	1.44	1.44	0.49	0.51
12	1.00	1.00	0.67	0.33

In conclusion, this section details how *Diamond Energy* II integrates ring substructures and oxygen atoms to expand the original algorithm's scope of applicability.

By differentiating between ring and non-ring atoms, the framework of the original algorithm is preserved and adapted to work effectively for ring systems.

Introducing oxygen necessitates the addition of new energy terms. An automated pipeline has been developed to facilitate the validation of its parameter set. Furthermore, a Boltzmann factor-based performance score has been established to aid in confirming the Bayesian optimization endpoint.

The subsequent section will assess the performance of *Diamond Energy* II.

3.3 *Diamond Energy* Further Development - Results and Conclusions

To test the performance of the current version of the program after the introduction of cyclohexyl and new atoms, comparative calculations were conducted between *Diamond Energy* II and four leading algorithms: SPMC, MCMM, LMCS, and MCLMCS. The test molecules, depicted in Figure 3.16 and Figure 3.17, were primarily chosen to evaluate different branching positions on the ring and the maximum number of rings they can incorporate, which represents their limit.

Diamond Energy II was tested by generating initial molecular structures in 3D .sdf format. These structures were then processed in two parallel methods: one involved adding hydrogen atoms using RDKit and then converting to .mae format; the other involved first converting the .sdf files to .mae format, followed by using MacroModel's applyhtreat function to add hydrogen. After these parallel processing steps, the structures were used as input files in MacroModel for the minimization process.

This minimization used the Full-Matrix Newton-Raphson Technique (FMNR) and continued until the gradient values were less than 0.001 kcal/Å-mol. After minimization, the vibrational frequencies was checked to ensure the structures were at true energy minima.

This process was compared with four established conformational search algorithms: MCMM, SPMC, LMCS, and MCLMCS, resulting in six different test pipelines: Diamond_RDKit, Di-

amond_MacroModel, MCMM, SPMC, LMCS, and MCLMCS. To ensure a fair comparison, all algorithms were tested using two different force fields: MM3 and OPLS4.

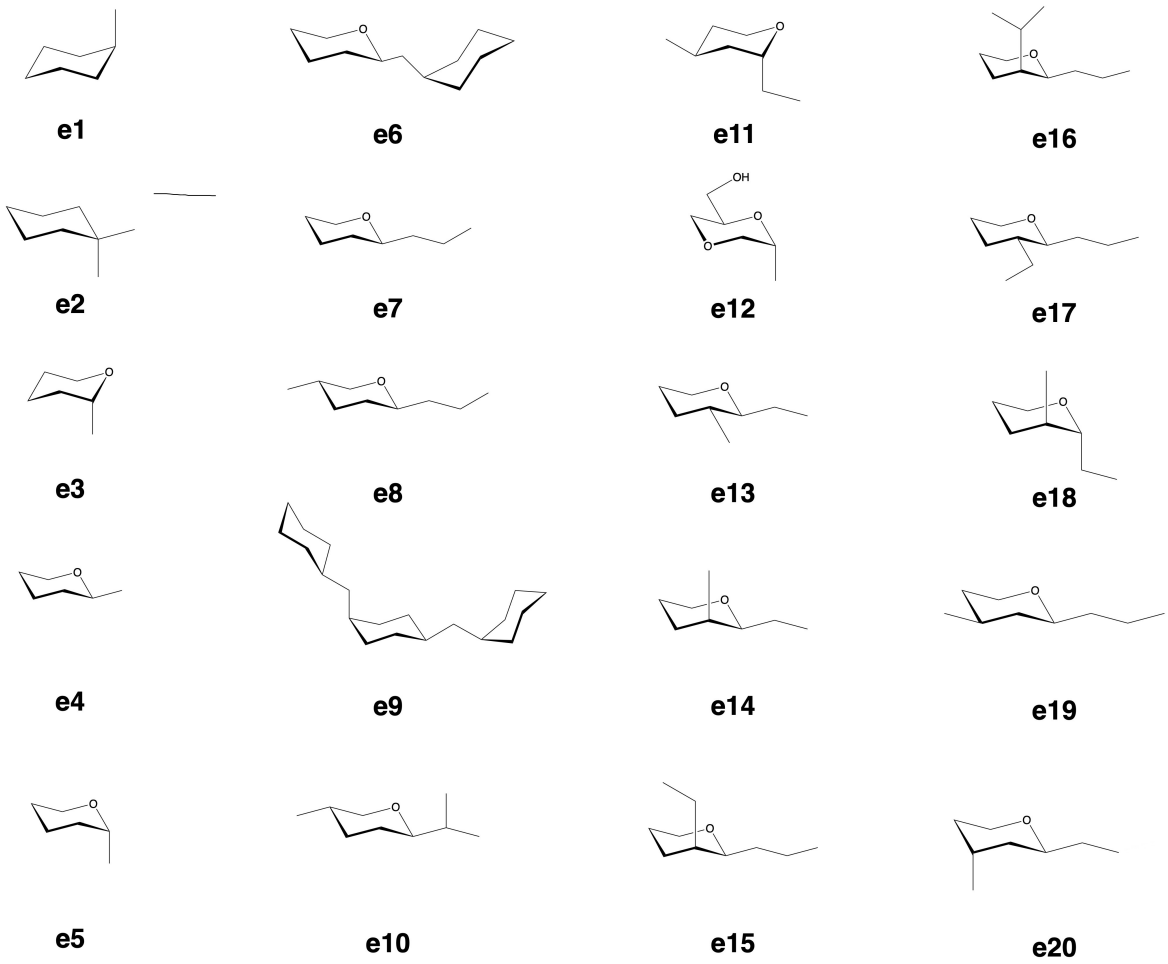


Figure 3.16 Molecules used for testing the performance of *Diamond Energy* II Further development.

For a detailed illustration of the comparison process and the calculations involved, please see Figure 3.18.

3.3.1 A Detailed Analysis and Illustration of *Diamond Energy* II Results

To demonstrate the mechanism of *Diamond Energy* II, the test molecule **e4** will be used as an example to walk through the entire process step by step. (Figure 3.19)

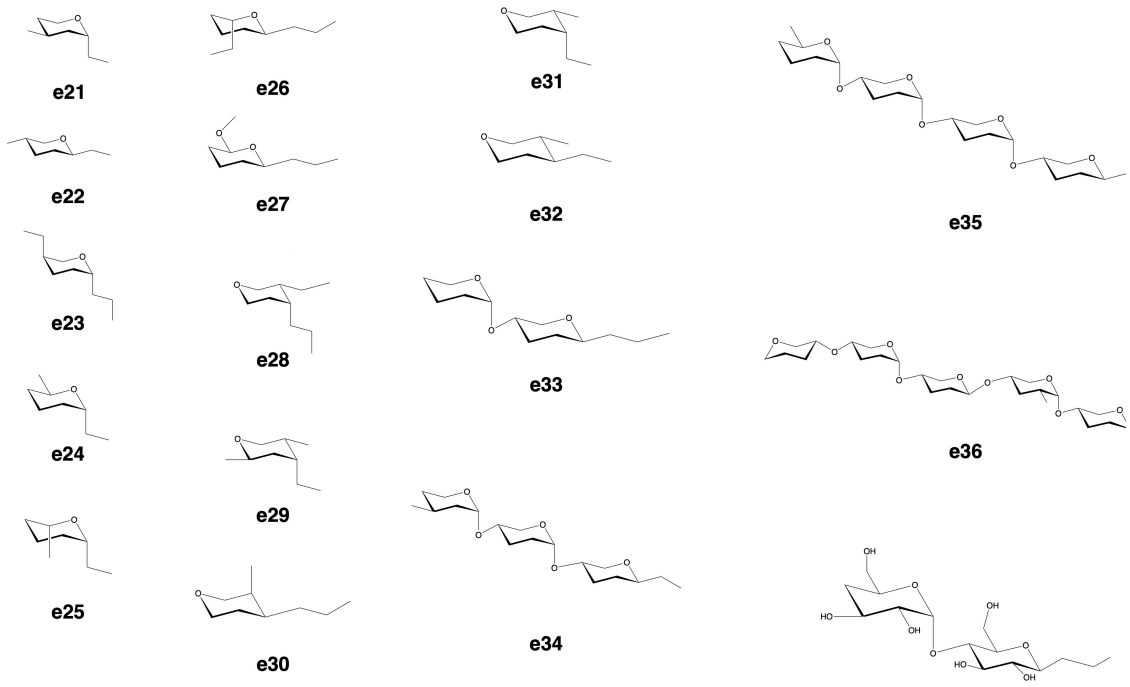


Figure 3.17 Molecules used for testing the performance of *Diamond Energy* II Further development.

Similar to *Diamond Energy* I, the program accepts an **e4** InChI string as input. The treatment of non-ring atoms remains unchanged from the original version, and therefore will not be reiterated here. The primary update in *Diamond Energy* II concerns the handling of ring atoms. For the molecule **e4**, the program initiates distinct protocols when encountering ring atoms, specifically atoms numbered 6, 4, 2, 3, 5, 7 (as per InChI notation). The sequence in which these atoms are processed is indicated by red arrows in the illustration provided below (Figure 3.20).

In its default setting, the program constructs the ring in a beta configuration. However, if subsequent stereochemical checks indicate an incorrect configuration, the program will then adjust the ring to an alpha configuration. This way to reconfigure the ring structure in *Diamond Energy* II is one of the changes made compared with its predecessor, *Diamond Energy* I.

In this instance, as the ring is correctly constructed, the molecule will proceed to the conformer generation phase and subsequently to the energy evaluation stage.

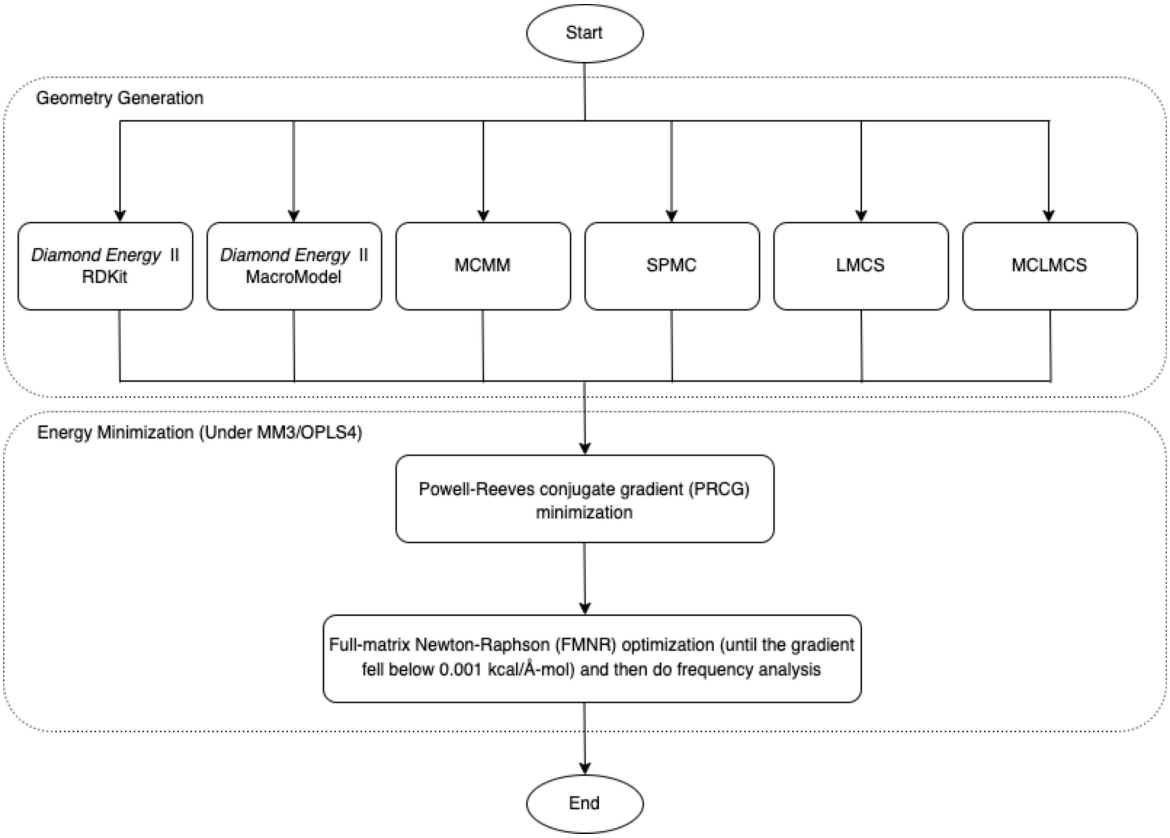
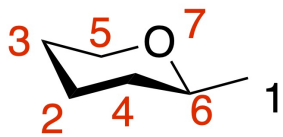


Figure 3.18 Comparative Workflow of *Diamond Energy* Version II and Four Leading Conformational Search Algorithms with Energy Minimization and Frequency Analysis.

e4

InChI=1S/C6H12O/c1-6-4-2-3-5-7-6/h6H,2-5H2,1H3/t6-/m0/s1

Figure 3.19 Molecule **e4** example.



e4

InChI=1S/C6H12O/c1-6-4-2-3-5-7-6/h6H,2-5H2,1H3/t6-/m0/s1

Figure 3.20 Molecule **e4** example - *Diamond Energy* II read the information of its input process.

Once more, the conformer generation process for alkane atoms remains unchanged; for further information, please refer to the previous sections, which will not be reiterated here. The key difference in handling ring structures is that instead of rotating rotatable bonds as with alkanes to create new geometries, the program flips the ring within the diamond lattice (Figure 3.21). This flip is precalculated and the corresponding coordinates for the new ring configuration are stored, allowing for rapid retrieval and generation of the new geometry, in a manner similar to that of nonring atoms.

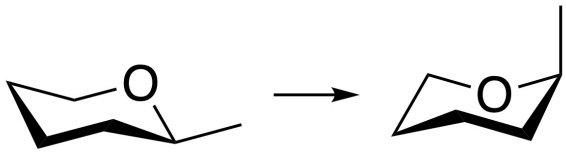


Figure 3.21 Molecule **e4** example - *Diamond Energy* II flip the ring feature.

In the energy evaluation phase, the process remains consistent with the previous methodology, with the addition of new parameters to account for the presence of oxygen.

Ultimately, using the **e4** molecule as an example, *Diamond Energy* II will output conformers following MM3 calculations, which are comparable in quality to those produced by the leading algorithm, MCLMCS (Table 3.5). Both methods identified the axial and equatorial six member ring configurations as low energy conformers.

Table 3.5 **e4** example - low energy conformers identified of *Diamond Energy* II method and MCLMCS method, using MM3 calculations (kcal/mol).

Conformer	*Diamond Energy* II	MCLMCS
1	9.59	9.59
2	9.59	12.41
3	12.41	15.28

3.3.2 A Comparative Analysis the Accuracy of *Diamond Energy* II

3.3.2.1 Identifying the Lowest Energy in the Test Set Using *Diamond Energy* II

The accuracy of *Diamond Energy* II is assessed by determining its ability to identify the lowest energy conformer and explore the low energy space, in comparison with other methods. Following the completion of experiments, the final output log files were analyzed. These log

files generated by MacroModel calculation automatically, having undergone a frequency test, were processed through a Python script to extract the true minimum energy values, excluding any saddle points and structures that failed to converge. The subsequent paragraphs detail these results.

Table 3.6 displays the lowest energy values for test molecules obtained using six different pipelines using MM3: Diamond_RDKit, Diamond_MacroModel, MCMM, SPMC, LMCS, and MCLMCS. This comparison shows that Diamond Energy II, whether following the RDKit or MacroModel process for adding hydrogen, consistently identifies low energy structures similar to those found by the four leading methods. However, this observation is preliminary and based on visual inspection. The subsequent paragraphs will provide a statistical analysis to yield more precise results.

To evaluate the performance of the six pipelines in identifying the lowest energy conformer for all test molecules, statistical calculations were performed using a Python script and its statis package. These calculations included determining the mean lowest energy for the entire test set for each pipeline, as well as the overall standard deviation, minimum and maximum values for all test molecules, and the 25th, 50th (median), and 75th percentiles for the lowest total energy achieved by each method. Additionally, the script counted the number of times each method found the lowest energy value for the same test molecule among the six pipelines, as shown in Table 3.7. The results indicate that the values for all six methods are relatively similar.

To visually represent the lowest energy values for the entire set of test molecules for each method, both a boxplot and a scatter plot were created.

A boxplot was constructed to display the distribution of the lowest energy values obtained by various methods for a set of test molecules. Displayed in Figure 3.22, this boxplot compares the performance of six different pipelines, all employing the MM3 force field. The *Diamond Energy* algorithm, applied through the RDKit pipeline (the first box from the left), shows results comparable to the leading algorithms - MCMM, MCLMCS, LMCS, SPMC (counted from the right). However, the box representing the MacroModel, which adds hydrogen to the *Diamond Energy* algorithm's output (the second box from the left), demonstrates less competitive results. This again suggests that the potential energy surface (PES) of the test molecules is complex and that the addition of hydrogen to the conformer search output can significantly impact the effectiveness of PES optimization, resulting in varied optimized conformers.

Table 3.6 Comparison of Lowest Energy Values Across Test Molecules Using Six Pipelines. This table presents a side-by-side comparison of the lowest energy values obtained for each test molecule by six different pipelines: Diamond_RDKit, Diamond_MacroModel, MCMM, SPMC, LMCS, and MCLMCS, all utilizing the MM3 Force Field (kcal/mol). The first column lists the names of the test molecules, while the subsequent columns display the lowest energy values identified by each of the six methods. These values are the result of a comprehensive conformer search and minimization process conducted in MacroModel.

Molecule	Diamond_MM3_MacroModel	Diamond_MM3_RDKit	SPMC_MM3	MCMM_MM3	MCLMCS_MM3	LMCS_MM3
e1	9.10	9.10	9.10	9.10	9.10	9.10
e2	12.29	12.29	12.29	12.29	12.29	12.29
e3	9.59	9.59	9.59	9.59	9.59	9.59
e4	9.59	9.59	9.59	9.59	9.59	9.59
e5	9.59	9.59	9.59	9.59	9.59	9.59
e6	20.04	20.04	20.04	20.04	20.04	20.04
e7	12.04	12.04	12.04	12.04	12.04	12.04
e8	13.01	13.01	13.01	13.01	13.01	13.01
e9	32.37	32.37	39.43	32.37	32.37	32.37
e10	14.41	14.41	14.41	14.41	14.41	14.41
e11	13.65	13.65	13.65	13.65	13.65	13.65
e12	43.26	42.81	42.81	42.81	42.81	42.81
e13	13.22	13.22	13.22	13.22	13.22	13.22
e14	14.54	14.54	14.54	14.54	14.54	14.54
e15	18.02	18.02	18.02	18.02	18.02	18.02
e16	22.15	22.15	22.15	22.15	22.15	23.98
e17	16.64	16.64	16.64	16.64	16.64	16.64
e18	13.22	13.22	13.22	13.22	13.22	13.22
e19	12.94	12.94	12.94	12.94	12.94	12.94
e20	13.65	13.65	13.65	13.65	13.65	13.65
e21	13.65	13.65	13.65	13.65	13.65	13.65
e22	11.96	11.96	11.96	11.96	11.96	11.96
e23	14.98	14.98	14.98	14.98	14.98	14.98
e24	14.92	14.92	14.92	14.92	14.92	14.92
e25	12.08	12.08	12.08	12.08	12.08	12.08
e26	17.18	17.18	17.18	17.18	17.18	17.18
e27	2.51	2.51	2.51	2.51	2.51	2.51
e28	18.26	18.26	17.36	17.36	17.36	17.36
e29	16.95	16.95	16.95	16.95	16.95	16.95
e30	14.84	14.84	14.84	14.84	14.84	14.84
e31	14.84	14.84	14.84	14.84	14.84	14.84
e32	13.71	13.71	13.71	13.71	13.71	13.71
e33	20.81	20.81	22.55	20.81	20.81	20.81
e34	29.61	29.61	34.51	29.61	29.61	29.61
e35	46.57	43.13	46.10	38.65	38.65	43.13
e36	82.10	73.59	83.25	75.02	75.42	75.86
e37	121.38	121.24	126.69	120.44	120.44	120.44

Table 3.7 This table presents statistics including the mean, standard deviation, minimum, 25th percentile, median (50th percentile), 75th percentile, and maximum values for the 'Lowest_Total_Energy' achieved by each method. Additionally, a column titled 'Lowest Energy Found Times' is included, which indicates the frequency with which each method identified the lowest energy value for a specific molecule in the test set. This table offers a comprehensive view of the central tendency and variability of the energy values obtained by each method using MM3.

Method	Mean	STD	Min	25%	50%	75%	Max	Lowest Energy Found Times
Diamond_MM3_MacroModel	21.34	21.81	2.51	12.94	14.54	18.26	121.38	32
Diamond_MM3_RDKit	21.00	21.04	2.51	12.94	14.54	18.26	121.24	34
LMCS_MM3	21.07	21.11	2.51	12.94	14.54	18.02	120.44	34
MCLMCS_MM3	20.88	20.95	2.51	12.94	14.54	18.02	120.44	36
MCMM_MM3	20.87	20.92	2.51	12.94	14.54	18.02	120.44	36
SPMC_MM3	21.84	22.74	2.51	12.94	14.54	18.02	126.69	31

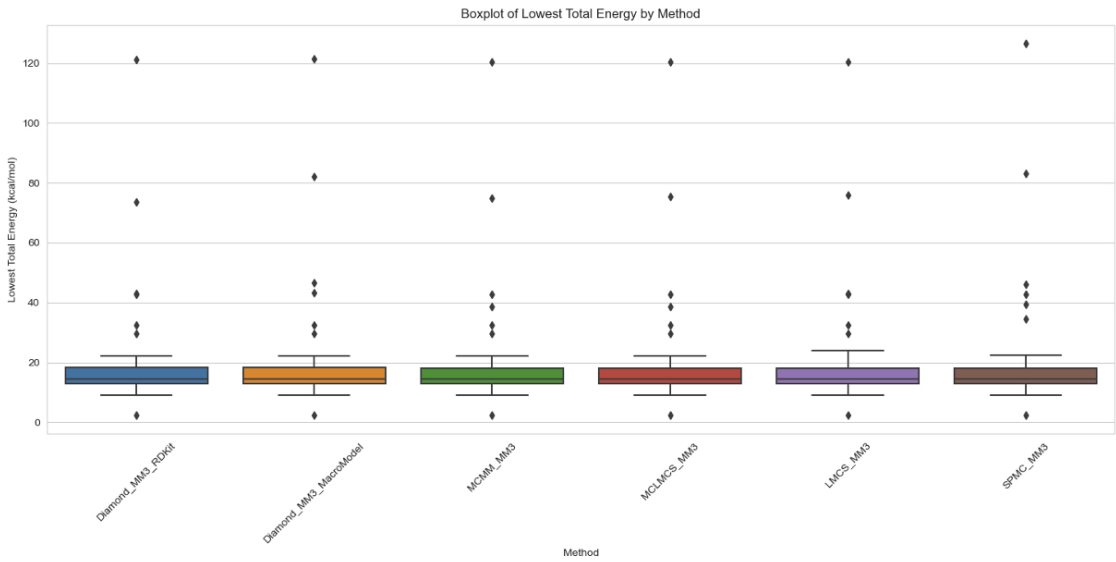


Figure 3.22 A boxplot illustrating the distribution of lowest energy values for all molecules in the test set, across the six pipelines, using the MM3 force field.

The image depicts a scatter plot in Figure 3.23, which illustrates the minimum energy values for a collection of molecules tested using six different pipelines, all employing the MM3 force field. The plot includes red and blue dots, representing the geometry search results from *Diamond Energy* II using the MacroModel and RDKit package, respectively, with hydrogen added independently. The dots from other methodologies are displayed in various colors. There is a noticeable overlap among the dots, indicating that the six pipelines generally yield comparable results for these molecules. However, for molecules **e35** and **e36**, the results from the Diamond methods appear to be less effective at identifying the lowest energy states in comparison to the other techniques. This discrepancy may be attributed to the fact that for molecules with more than four sugar units, the conformers with the lowest energy may not align well within a diamond lattice structure, suggesting that these molecules may exceed the optimal application range for this algorithm.

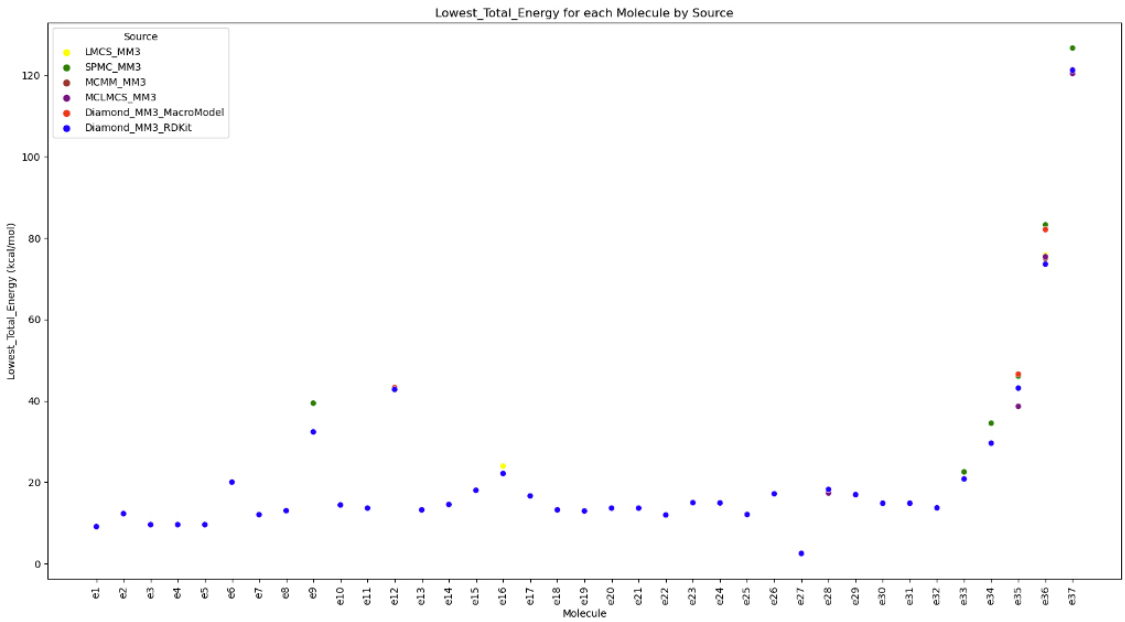


Figure 3.23 A scatter plot that showcases the minimum energy levels for each molecule in the test set, as obtained by all six pipelines using the MM3 force field.

To quantify the closeness of these six methods, an ANOVA test was performed using the python stats package. The results are as follows.

F-statistic = 0.011: This calculated F-value from the ANOVA test represents a ratio of variances, used to determine if there are significant differences between the means of different groups. Typically, a higher F-statistic indicates a greater likelihood of significant differences

between the means. This low value of the test shows that there is no big gap between these groups results.

P-value = 0.999: In most studies, a p-value of less than 0.05 is considered statistically significant. In this case, the very high p-value (close to 1) suggests that the observed differences in means are likely due to random chance, indicating no statistically significant difference between the groups.

Our results show that there are no statistically significant differences between the methods regarding the 'Lowest_Total_Energy' they find. The statistical analysis suggests that all six methods perform similarly in identifying the lowest total energy for the tested molecules, which implies that the *Diamond Energy* II extension algorithm has comparable accuracy in conformer search to the four leading algorithms under MM3 force field: MCMM, LMCS, MCLMCS, and SPMC.

Similarly, using the OPLS4 force field yielded comparable results. These findings are detailed and analyzed in the paragraphs that follow. Table 3.8 presents the lowest energy values for each test molecule across all methods. As observed with the MM3 force field, *Diamond Energy* II also demonstrated competent performance in identifying the lowest energy conformers when using OPLS4, comparable to the other methods.

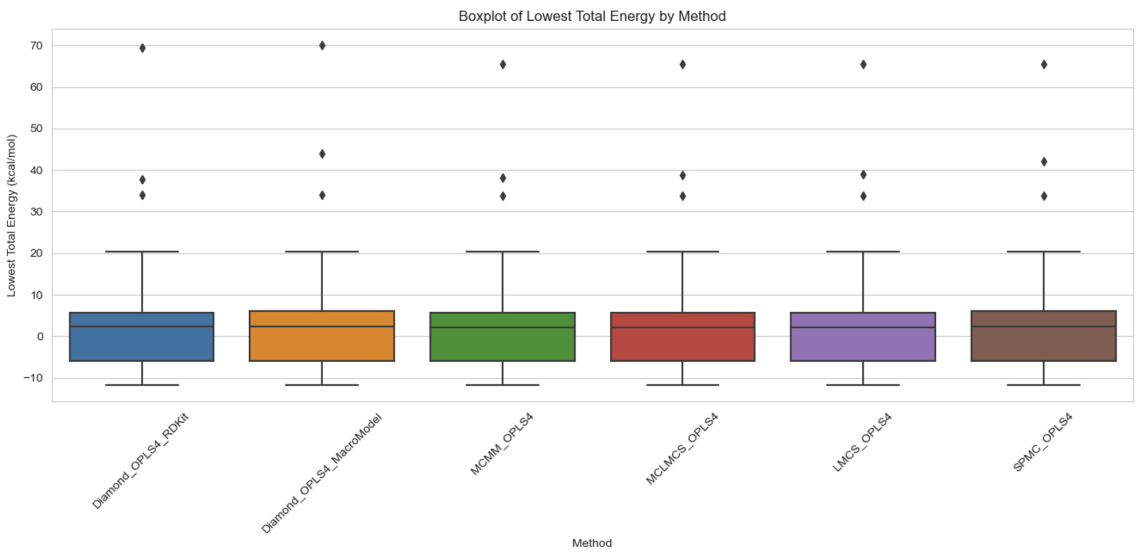


Figure 3.24 A boxplot illustrating the distribution of lowest energy values for all molecules in the test set, across the six pipelines, using the OPLS4 force field.

Statistical analysis was conducted using a Python script equipped with the 'statis' package to process data obtained by the OPLS4 force field calculations as well. This analysis included

Table 3.8 Comparison of Lowest Energy Values Across Test Molecules Using Six Pipelines. This table presents a side-by-side comparison of the lowest energy values obtained for each test molecule by six different pipelines: Diamond_RDKit, Diamond_MacroModel, MCMM, SPMC, LMCS, and MCLMCS, all utilizing the OPLS4 Force Field (kcal/mol). The first column lists the names of the test molecules, while the subsequent columns display the lowest energy values identified by each of the six methods. These values are the result of a comprehensive conformer search and minimization process conducted in MacroModel.

Molecule	Diamond_OPLS4_MacroModel	Diamond_OPLS4_RDKit	SPMC	MCMM	MCLMCS	LMCS
e1	5.66	5.66	5.66	5.66	5.66	5.66
e2	5.32	5.32	5.32	5.32	5.32	5.32
e3	−10.07	−10.07	−10.08	−10.08	−10.08	−10.08
e4	−10.07	−10.07	−10.07	−10.07	−10.07	−10.07
e5	−10.07	−10.07	−10.08	−10.08	−10.08	−10.08
e6	1.34	1.34	1.34	1.34	1.34	1.34
e7	−1.88	−1.88	−1.88	−1.88	−1.88	−1.88
e8	1.81	1.81	1.81	1.81	1.81	1.81
e9	20.40	20.40	20.40	20.40	20.40	20.40
e10	7.44	7.44	7.44	7.44	7.44	7.44
e11	−5.93	−5.93	−5.93	−5.93	−5.93	−5.93
e12	33.94	33.94	33.78	33.78	33.78	33.78
e13	5.41	5.41	5.41	5.41	5.41	5.41
e14	8.29	8.29	8.29	8.29	8.29	8.29
e15	7.93	7.93	7.94	7.94	7.94	7.94
e16	6.07	6.07	6.07	6.07	6.07	6.07
e17	6.33	6.33	6.32	6.32	6.32	6.32
e18	5.41	5.41	5.41	5.41	5.41	5.41
e19	−7.07	−7.07	−7.08	−7.08	−7.08	−7.08
e20	−5.93	−5.93	−5.94	−5.94	−5.94	−5.94
e21	−5.93	−5.93	−5.93	−5.93	−5.93	−5.93
e22	2.10	2.10	2.10	2.10	2.10	2.10
e23	1.36	1.36	1.37	1.37	1.37	1.37
e24	−8.18	−8.18	−8.18	−8.18	−8.18	−8.18
e25	−11.70	−11.70	−11.70	−11.70	−11.70	−11.70
e26	−1.61	−1.61	1.19	−1.61	−1.61	−1.61
e27	−10.92	−10.92	−10.92	−10.92	−10.92	−10.92
e28	4.16	4.16	5.41	5.41	5.41	5.41
e29	−5.90	−5.90	−5.90	−5.90	−5.90	−5.90
e30	2.36	2.36	2.36	2.36	2.36	2.36
e31	2.36	2.36	2.36	2.36	2.36	2.36
e32	3.59	3.59	3.59	3.59	3.59	3.59
e33	4.90	4.90	6.17	4.90	4.90	4.90
e34	0.44	0.44	0.44	0.44	0.44	0.44
e35	7.12	4.49	5.35	0.15	0.15	2.17
e36	44.05	37.81	42.16	38.13	38.72	39.07
e37	70.00	69.37	65.56	65.56	65.56	65.56

Table 3.9 This table presents statistics including the mean, standard deviation, minimum, 25th percentile, median (50th percentile), 75th percentile, and maximum values for the 'Lowest_Total_Energy' achieved by each method. Additionally, a column titled 'Lowest Energy Found Times' is included, which indicates the frequency with which each method identified the lowest energy value for a specific molecule in the test set. This table offers a comprehensive view of the central tendency and variability of the energy values obtained by each method using OPLS4.

Method	Mean	STD	Min	25%	50%	75%	Max	Lowest Energy Found Times
Diamond_OPLS4_RDKit	4.14	15.35	−11.70	−5.93	2.36	5.66	69.37	24
Diamond_OPLS4_MacroModel	4.39	15.84	−11.70	−5.93	2.36	6.07	70.00	23
LMCS_OPLS4	4.03	14.98	−11.70	−5.93	2.17	5.66	65.56	24
MCLMCS_OPLS4	3.97	14.97	−11.70	−5.93	2.10	5.66	65.56	25
MCMM_OPLS4	3.95	14.93	−11.70	−5.93	2.10	5.66	65.56	25
SPMC_OPLS4	4.31	15.17	−11.70	−5.93	2.36	6.07	65.56	23

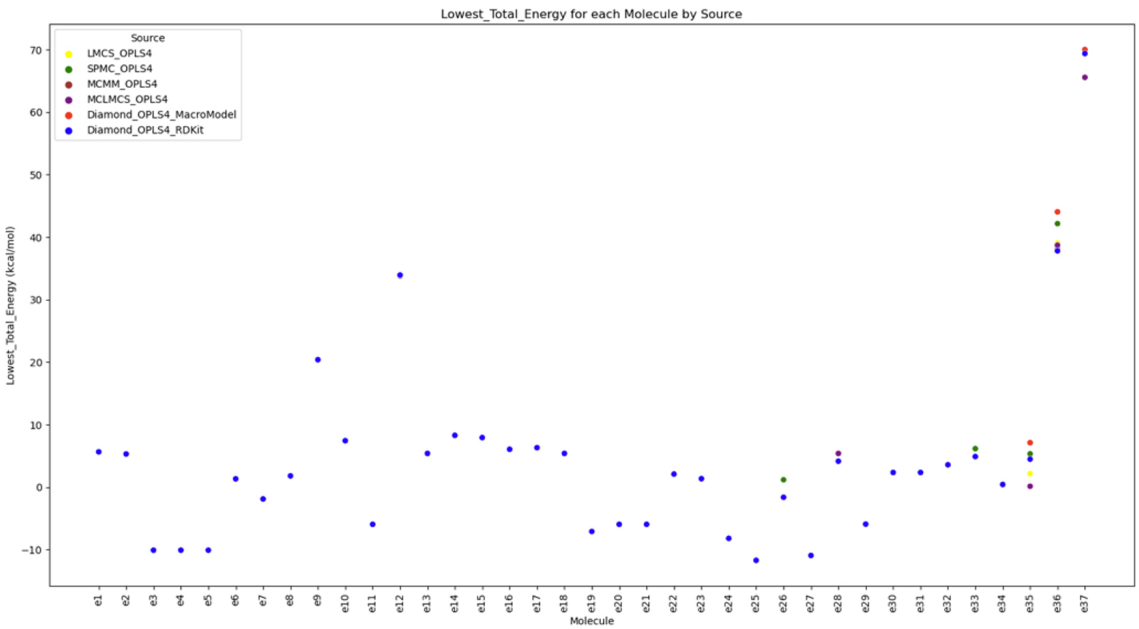


Figure 3.25 Displays a scatter plot that showcases the minimum energy levels for each molecule in the test set, as obtained by all six pipelines using the OPLS4 force field.

the calculation of the mean lowest energy per pipeline for the test set, along with the standard deviation, minimum, maximum, and percentile values (25th, 50th, and 75th) for the lowest total energy achieved by each method. The frequency of each method obtaining the lowest energy for identical test molecules across six pipelines was also recorded. Results, detailed in Table 3.9, showed negligible differences in these statistical measures across all methods, indicating a consistent performance among the six tested methods under the OPLS4 force field.

This section compares the distribution of the lowest energy values calculated by all methods for a given test set, with the calculations performed using the OPLS4 force field. Figure 3.24 features a box plot that aggregates and displays the range of energy values for all molecules in the test set, derived from six different pipelines using OPLS4. In parallel, Figure 3.25 provides a scatter plot that pinpoints the individual lowest energy values for each molecule within the same set and across the same six pipelines, also employing OPLS4. The data visualized in both the box plot and scatter plot reveal a consistent distribution pattern of the lowest energy values when transitioning from the MM3 to the OPLS4 force field.

To assess the consistency of the six methods in determining the lowest energy value across the entire dataset with the OPLS4 force field, an ANOVA test was performed on the energy values obtained, similar to the analysis done for MM3's results. The ANOVA test outcomes show no statistically significant variance in the lowest energy values across the methods. An F-statistic of roughly 0.0054 and a p-value near 1.00, significantly above the standard alpha level of 0.05, suggesting that any differences in group means are likely due to random variation rather than actual differences between the methods.

In conclusion, the analysis of the lowest energy values identified for the entire test set by six methods both using MM3 and OPLS4, *Diamond Energy* II with both MacroModel and RDKit post-processing, MCMM, MCLMCS, LMCS, and SPMC has been comprehensive. Statistical data including mean, minimum, maximum, and quartile values, along with box plots and scatter plots, as well as results from the ANOVA test, all indicate that the *Diamond Energy* II algorithm is as effective as the other four leading algorithms in searching for conformers.

3.3.2.2 Determining Conformers Within 3 to 5 kcal/mol of the Minimum Energy Using *Diamond Energy* II in the Test Set

The lowest energy found in a conformer search method can serve as a criterion for evaluating its accuracy. Additionally, the count of conformers identified within energy ranges of 3 kcal/mol and 5 kcal/mol from the lowest energy provides insight into the method's effectiveness.

Table 3.10 below displays the number of conformers identified within a 3 kcal/mol range from the lowest energy level for six pipelines: Diamond_RDKit, Diamond_MacroModel, SPMC, MCMM, MCLMCS, and LMCS, all calculated using the MM3 force field.

In Table 3.10, the first column lists the test molecule numbers. The second and third columns show the results from *Diamond Energy* II, with RDKit and MacroModel applied respectively for adding hydrogens prior to minimization. The subsequent columns contain data from the four other leading methods. As indicated in Table 3.10, *Diamond Energy* II consistently identifies more conformers within a 3 kcal/mol range of the lowest energy level in most test cases.

To evaluate the average number of conformers identified across all test cases for each method, the mean number of conformers found by each method across the entire dataset were calculated. These averages are presented in Table 3.11 showing that, on average, Diamond_RDKit identifies the most conformers, followed by Diamond_MacroModel, SPMC, LMCS, MCMM, and MCLMCS. This suggests that *Diamond Energy* II is particularly effective in exploring low-energy conformations, as indicated by its ability to identify conformers within a 3 kcal/mol range of the lowest energy. Furthermore, the systematic search methods, which include the first two *Diamond Energy* II pipelines, SPMC, and LMCS, demonstrate a superior performance in approaching the global minimum, as reflected by their higher mean values in Table 3.11.

Table 3.12 shows the count of conformers identified within a 5 kcal/mol range from the lowest energy level for the six pipelines: Diamond_RDKit, Diamond_MacroModel, SPMC, MCMM, MCLMCS, and LMCS, each utilizing the MM3 force field for calculations. In most test cases, as illustrated in Table 3.12, the two *Diamond Energy* II pipelines identified a greater number of conformers compared to the other methods.

The average number of conformers identified by each method for the entire dataset was again computed (Table 3.13). On average, Diamond_RDKit finds the most conformers,

Table 3.10 Conformers Identified Within a 3 kcal/mol Range from the Lowest Energy Using MM3 Force Field Calculations for Six Computational Pipelines.

Molecule	Diamond_MM3_MacroModel	Diamond_MM3_RDKit	SPMC	MCMM	MCLMCS	LMCS
e1	3	3	2	2	2	2
e2	3	3	1	1	1	1
e3	3	3	2	2	2	2
e4	3	3	2	2	2	2
e5	3	3	2	2	2	2
e6	19	19	11	11	10	10
e7	15	15	8	8	8	7
e8	7	7	7	7	7	7
e9	36	36	28	14	11	11
e10	6	6	3	3	3	3
e11	8	8	5	5	5	5
e12	9	9	11	11	11	11
e13	6	6	3	3	3	3
e14	5	5	3	3	3	3
e15	25	25	15	15	15	14
e16	14	14	10	10	10	14
e17	28	28	14	14	14	14
e18	6	6	3	3	3	3
e19	14	14	7	7	7	7
e20	8	8	5	5	5	5
e21	8	8	5	5	5	5
e22	6	6	3	3	3	3
e23	40	40	20	20	20	20
e24	8	8	5	5	5	5
e25	6	6	3	3	3	3
e26	41	41	28	28	28	28
e27	20	20	10	10	10	10
e28	24	24	22	22	22	22
e29	6	6	4	4	4	4
e30	6	6	4	4	4	4
e31	6	6	4	4	4	4
e32	8	8	5	5	5	5
e33	76	76	16	19	19	19
e34	48	48	23	11	12	12
e35	52	48	29	21	22	33
e36	272	8	84	38	24	33
e37	30	90	32	40	52	55

Table 3.11 Average Number of Conformers (Within a 3 kcal/mol Range of the Lowest Energy) Identified by Each Method Using MM3 Force Field Calculations.

Method	Mean
Diamond_RDKit	24
Diamond_MacroModel	18
SPMC	12
LMCS	11
MCMM	10
MCLMCS	10

with Diamond_MacroModel, SPMC, LMCS, MCMM, and MCLMCS following in that order. The data suggest that *Diamond Energy* II excels in identifying low-energy conformers, within a 5 kcal/mol range of the lowest energy. Additionally, the systematic search methods — *Diamond Energy* II's first two pipelines, along with SPMC and LMCS — show better performance in approaching the global minimum, which is evident from their higher average conformer counts in Table 3.13.

The same analysis process was applied to results obtained using the OPLS4 force field calculations. The next tables shows the number of conformers found within a 3 kcal/mol and 5 kcal/mol range from the lowest energy for six pipelines: Diamond_RDKit, Diamond_MacroModel, SPMC, MCMM, MCLMCS, and LMCS, all calculated with the OPLS4 force field. Both Table 3.14 and Table 3.15 indicate that, in most cases, the *Diamond Energy* II pipelines found a higher number of conformers than the other methods.

Table 3.16 and Table 3.17 display the mean number of conformers identified by each method across the dataset, for both conformers within a 3 kcal/mol and 5 kcal/mol range of the lowest energy. Similarly, it shows that *Diamond Energy* II, on average, identifies more low-energy conformers both within a 3 kcal/mol and 5 kcal/mol range of the lowest energy using OPLS4 force field calculations.

Conclusions The results and analysis show that *Diamond Energy* II, along with SPMC, LMCS, MCMM, and MCLMCS, are effective conformer search methods when using MM3 and OPLS4 force fields. They have successfully identified the lowest energy values in most sugar-like test cases. Furthermore, *Diamond Energy* II has proven to be competent by finding a greater number of low-energy conformers within 3 kcal/mol and 5 kcal/mol ranges from the lowest energy value, compared to the other four leading methods.

Table 3.12 Conformers Identified Within a 5 kcal/mol Range from the Lowest Energy Using MM3 Force Field Calculations for Six Computational Pipelines.

Molecule	Diamond_RDKit	Diamond_MacroModel	SPMC	MCMM	MCLMCS	LMCS
e1	3	3	2	2	2	2
e2	3	3	1	1	1	1
e3	3	3	2	2	2	2
e4	3	3	2	2	2	2
e5	3	3	2	2	2	2
e6	31	31	22	22	17	17
e7	23	23	16	16	16	11
e8	9	9	14	14	14	11
e9	104	104	98	47	37	37
e10	7	7	4	4	4	3
e11	8	8	8	8	8	8
e12	9	9	15	15	15	15
e13	8	8	5	5	5	5
e14	8	8	5	5	5	5
e15	46	46	29	29	29	25
e16	40	40	27	27	27	28
e17	52	52	32	32	32	32
e18	8	8	5	5	5	5
e19	18	18	11	11	11	11
e20	8	8	8	8	8	8
e21	8	8	8	8	8	8
e22	8	8	5	5	5	3
e23	58	58	37	37	37	33
e24	8	8	12	12	12	12
e25	6	6	3	3	3	3
e26	54	54	56	59	59	59
e27	41	41	24	24	24	24
e28	46	46	39	40	40	40
e29	7	7	5	5	5	5
e30	9	9	6	6	6	6
e31	9	9	6	6	6	6
e32	8	8	5	5	5	5
e33	185	185	42	52	49	49
e34	256	256	75	39	45	40
e35	192	136	72	43	61	116
e36	1169	20	262	101	62	71
e37	72	168	126	99	143	151

Table 3.13 Average Number of Conformers (Within a 5 kcal/mol Range of the Lowest Energy) Identified by Each Method Using MM3 Force Field Calculations.

Method	Mean
Diamond_RDKit	68
Diamond_MacroModel	38
SPMC	29
LMCS	23
MCLMCS	22
MCMM	22

3.3.2.3 A Comparative Analysis the Computational Efficiency of *Diamond Energy* II

To evaluate the speed of the *Diamond Energy* II conformer search, its running time will be compared with that of SPMC, LMCS, MCMM, and MCLMCS. The geometry generation time for *Diamond Energy* II is taken from its output log file, while its minimization time comes from the MacroModel minimization task log file. The total time for *Diamond Energy* II involves adding the geometry generation and minimization times together. For SPMC, LMCS, MCMM, and MCLMC, the running times are extracted from their respective MacroModel log files, as their processes are integrated within the MacroModel package. Their total time is the user time, which indicates the duration of the entire conformer search process. This is calculated by subtracting the minimization time, read directly from the log file, from the full round time.

Table 3.18 and Table 3.9 present two sets of data: the total time required to create geometries for test molecules and the average time needed to generate a single conformer for a specific geometry for all six pipelines.

Table 3.18 and Table 3.19, the first column lists the molecule names, and the second column shows the total and average time *Diamond Energy* II takes to generate conformers, separately. The subsequent columns provide the corresponding data for other methods. It is evident that in most test cases, *Diamond Energy* II requires significantly less time—less by several order of magnitude—to generate conformers (Figure 3.26 and Figure 3.27). This efficiency becomes more obvious with these sugar-like molecules, which are structurally more complex than the alkanes used in the previous *Diamond Energy* I version. In such complex systems, the efficiency advantages of *Diamond Energy* II's algorithm become apparent.

The upcoming content will detail the complete running time, including both conformer generation and minimization for *Diamond Energy* II compared to other methods using MM3

Table 3.14 Conformers Identified Within a 3 kcal/mol Range from the Lowest Energy Using OPLS4 Force Field Calculations for Six Computational Pipelines.

Molecule	Diamond_RDKit	Diamond_MacroModel	SPMC	MCMM	MCLMCS	LMCS
e1	3	3	2	2	2	2
e2	3	3	1	1	1	1
e3	3	3	2	2	2	2
e4	3	3	2	2	2	2
e5	3	3	2	2	2	2
e6	24	24	16	16	16	15
e7	19	19	12	12	12	12
e8	8	8	10	10	10	10
e9	40	40	19	26	19	19
e10	7	7	4	4	4	4
e11	8	8	5	5	5	5
e12	5	5	6	6	6	6
e13	7	7	4	4	4	3
e14	6	6	4	4	4	4
e15	24	24	17	17	17	17
e16	8	8	8	8	8	8
e17	36	36	19	19	19	19
e18	7	7	4	4	4	3
e19	16	16	9	9	9	9
e20	8	8	5	5	5	5
e21	8	8	5	5	5	5
e22	7	7	4	4	4	4
e23	44	44	24	24	24	24
e24	8	8	6	6	6	6
e25	6	6	3	3	3	3
e26	42	42	55	30	30	30
e27	18	18	10	10	10	10
e28	36	36	38	38	38	38
e29	7	7	5	5	5	5
e30	6	6	4	4	4	4
e31	6	6	4	4	4	4
e32	8	8	6	6	6	6
e33	136	136	20	30	30	30
e34	112	148	33	43	48	48
e35	220	120	37	75	119	123
e36	438	10	119	51	109	194
e37	72	134	28	45	56	56

Table 3.15 Conformers Identified Within a 5 kcal/mol Range from the Lowest Energy Using OPLS4 Force Field Calculations for Six Computational Pipelines.

Molecule	Diamond_RDKit	Diamond_MacroModel	SPMC	MCMM	MCLMCS	LMCS
e1	3	3	2	2	3	3
e2	3	3	1	1	1	1
e3	3	3	2	2	2	2
e4	3	3	2	2	2	2
e5	3	3	2	2	2	2
e6	39	39	32	32	32	26
e7	24	24	17	17	17	17
e8	9	9	18	18	18	18
e9	148	148	53	86	53	53
e10	7	7	9	9	9	9
e11	8	8	12	12	12	12
e12	7	7	10	10	10	10
e13	8	8	5	5	5	3
e14	8	8	6	6	6	6
e15	41	41	28	28	28	28
e16	25	25	21	21	21	21
e17	53	53	34	34	34	34
e18	8	8	5	5	5	3
e19	23	23	17	17	17	17
e20	8	8	12	12	12	12
e21	8	8	12	12	12	12
e22	8	8	7	7	7	7
e23	64	64	37	47	47	47
e24	9	9	12	12	12	12
e25	6	6	5	5	5	5
e26	57	57	78	84	82	84
e27	38	38	34	34	34	34
e28	48	48	63	65	65	65
e29	9	9	8	8	8	8
e30	9	9	6	6	6	6
e31	9	9	6	6	6	6
e32	9	9	11	11	11	11
e33	238	238	53	91	94	93
e34	397	490	91	140	183	171
e35	518	256	85	188	351	311
e36	2412	20	309	203	445	511
e37	190	270	67	119	175	183

Table 3.16 Average Number of Conformers (Within a 3 kcal/mol Range of the Lowest Energy) Identified by Each Method Using OPLS4 Force Field Calculations.

Method	**Mean**
Diamond_RDKit	38
Diamond_MacroModel	27
LMCS	20
MCLMCS	18
SPMC	15
MCMM	15

Table 3.17 Average Number of Conformers (Within a 5 kcal/mol Range of the Lowest Energy) Identified by Each Method Using OPLS4 Force Field Calculations.

Method	**Mean**
Diamond_RDKit	121
Diamond_MacroModel	53
LMCS	50
MCLMCS	50
MCMM	37
SPMC	32

and OPLS4 force field calculations. Tables 3.20 to Table 3.23 display the total and average complete running times for the six pipelines: Diamond_RDKit, Diamond_MacroModel, SPMC, MCMM, MCLMCS, and LMCS, using MM3 force field (Tables 3.20 and Table 3.21) and OPLS4 force field (Table 3.22 and Table 3.23).

Tables 3.20 and Table 3.21 list molecular names in the first column. The following two columns detail the complete total and average running times for the Diamond_MacroModel and Diamond_RDKit pipelines, respectively. The remaining columns in both tables provide the comparative data for the other four methods. The data reveal that for most sample molecules, Diamond_MacroModel and Diamond_RDKit pipelines typically take significantly less time than the other methods. This difference is visually represented in Figures 3.28 and Figure 3.29, which plot the complete total and average running times for conformer generation using MM3 calculations, with time in the y-axis seconds logarithmic scale for total running times, seconds on the y-axis for average running times and sample molecules on the x-axis. In these figures, the green and red bars represent Diamond_MacroModel and Diamond_RDKit, respectively, positioned in the middle of each molecular set, while the other methods are indicated by MCMM (blue), SPMC (yellow), MCLMCS (purple), and

Table 3.18 Total Running Time Cost of Each Method to Generate Conformers for All Test Cases.

Molecule	Diamond	MCMM	SPMC	MCLMCS	LMCS
e1	0.00354645	4.11	4.69	5.94	5.97
e2	0.00544216	4.96	5.6	6.48	6.95
e3	0.00575016	3.69	4.07	5.15	5.53
e4	0.00578379	4.22	3.71	4.6	5.68
e5	0.00588906	3.82	3.88	4.77	5.13
e6	0.02810545	8.01	65.41	11.02	11.41
e7	0.0149989	4.98	5.81	6.91	7.35
e8	0.01329733	5.29	5.97	6.78	7.18
e9	0.20022263	66.12	15.81	22.37	23.51
e10	0.00768406	4.78	4.97	6.89	7.74
e11	0.00925972	4.36	4.94	6.26	7.13
e12	0.00959489	3.33	4.11	5.2	5.38
e13	0.00825695	3.93	4.49	6.15	6.54
e14	0.00745436	3.95	4.28	5.97	6.76
e15	0.02946547	5.14	8.26	7.91	8.67
e16	0.03008113	5.8	9.91	9.12	10.07
e17	0.03059052	5.13	8.72	8.01	9.06
e18	0.00927765	3.83	4.48	6.09	6.84
e19	0.01389322	4.71	4.85	7.49	8.13
e20	0.00766447	4.27	4.6	6.72	7.26
e21	0.00929571	3.87	4.61	7.17	7.25
e22	0.00833213	4.25	4.17	6.6	7.94
e23	0.03195332	5.48	7.53	9.13	9.78
e24	0.01235577	4.26	4.65	7.24	7.45
e25	0.0074611	4.12	4.5	6.61	8.16
e26	0.0323145	6.42	7.48	9.66	10.5
e27	0.03209184	4.51	7.84	8.12	8.95
e28	0.0304047	5.6	7.25	9.02	10.47
e29	0.01004652	4.68	5.05	8.21	8.3
e30	0.00773047	4.44	4.69	7.49	7.62
e31	0.00959863	4.54	4.73	6.65	7.82
e32	0.007739	4.59	4.58	7.4	8.29
e33	0.24740509	11.74	8.94	13.5	14.39
e34	1.50956419	71.19	12.25	20.94	22.44
e35	27.0700318	364.5	17.84	33.31	34.41
e36	502.308096	866.96	28.41	43.73	44.03
e37	0.9909376	19.24	12.96	19.9	21.77

Table 3.19 Average Running Time Cost of Each Method to Generate Conformers for All Test Cases.

Molecule	Diamond	MCMM	SPMC	MCLMCS	LMCS
e1	0.001	2.055	2.345	2.970	2.985
e2	0.002	1.653	1.867	2.160	2.317
e3	0.002	1.230	1.357	1.717	1.843
e4	0.002	1.407	1.237	1.533	1.893
e5	0.002	1.273	1.293	1.590	1.710
e6	0.001	0.243	2.110	0.459	0.475
e7	0.001	0.262	0.306	0.364	0.565
e8	0.001	0.331	0.373	0.424	0.653
e9	0.001	0.735	0.121	0.320	0.341
e10	0.001	1.195	1.243	1.723	2.580
e11	0.001	0.335	0.380	0.482	0.548
e12	0.001	0.208	0.257	0.325	0.336
e13	0.001	0.655	0.748	1.025	1.090
e14	0.001	0.564	0.611	0.853	0.966
e15	0.000	0.143	0.229	0.220	0.271
e16	0.001	0.161	0.275	0.253	0.288
e17	0.000	0.139	0.236	0.216	0.245
e18	0.001	0.638	0.747	1.015	1.140
e19	0.001	0.428	0.441	0.681	0.739
e20	0.001	0.328	0.354	0.517	0.558
e21	0.001	0.298	0.355	0.552	0.558
e22	0.001	0.850	0.834	1.320	2.647
e23	0.000	0.127	0.184	0.212	0.296
e24	0.001	0.304	0.332	0.517	0.532
e25	0.001	1.373	1.500	2.203	2.720
e26	0.000	0.074	0.090	0.111	0.121
e27	0.000	0.133	0.245	0.239	0.263
e28	0.001	0.117	0.154	0.188	0.218
e29	0.001	0.468	0.505	0.821	0.830
e30	0.001	0.493	0.521	0.832	0.847
e31	0.001	0.504	0.526	0.739	0.869
e32	0.001	0.656	0.654	1.057	1.184
e33	0.000	0.154	0.142	0.208	0.215
e34	0.000	1.249	0.114	0.299	0.387
e35	0.000	4.796	0.198	0.374	0.210
e36	0.001	9.223	0.122	0.717	1.001
e37	0.001	0.141	0.065	0.107	0.112

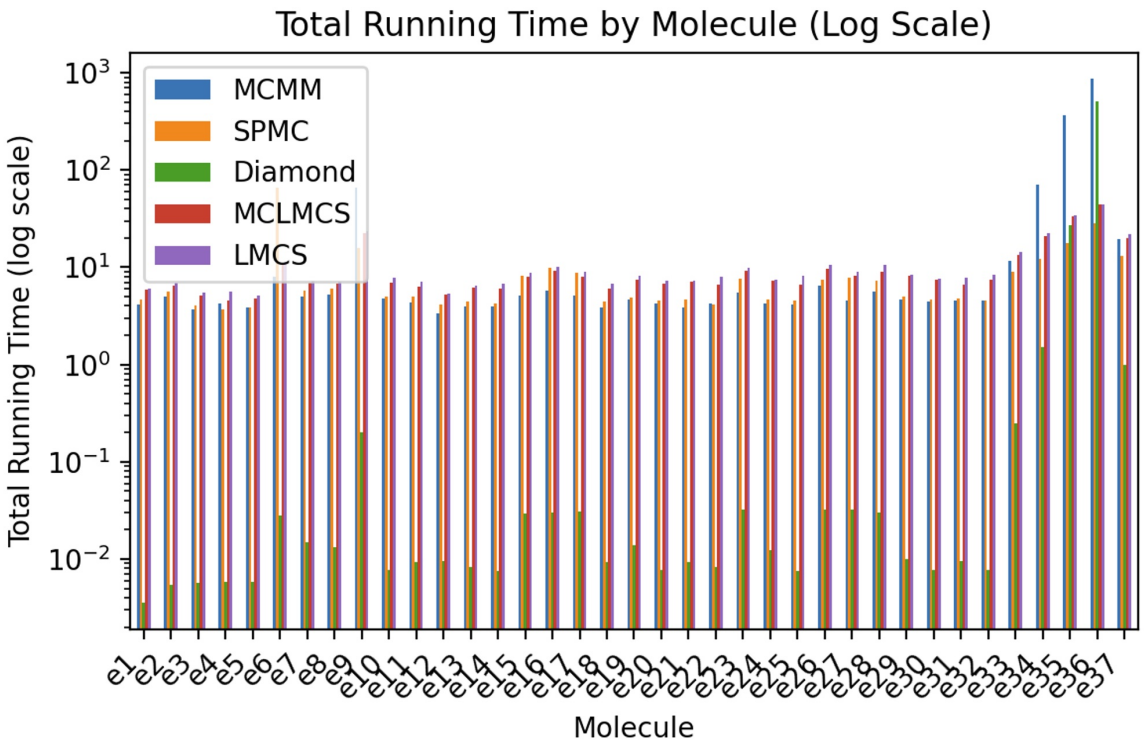


Figure 3.26 The total time cost to generate conformers of various methods: *Diamond Energy* II, SPMC, MCMM, LMCS, MCLMCS.

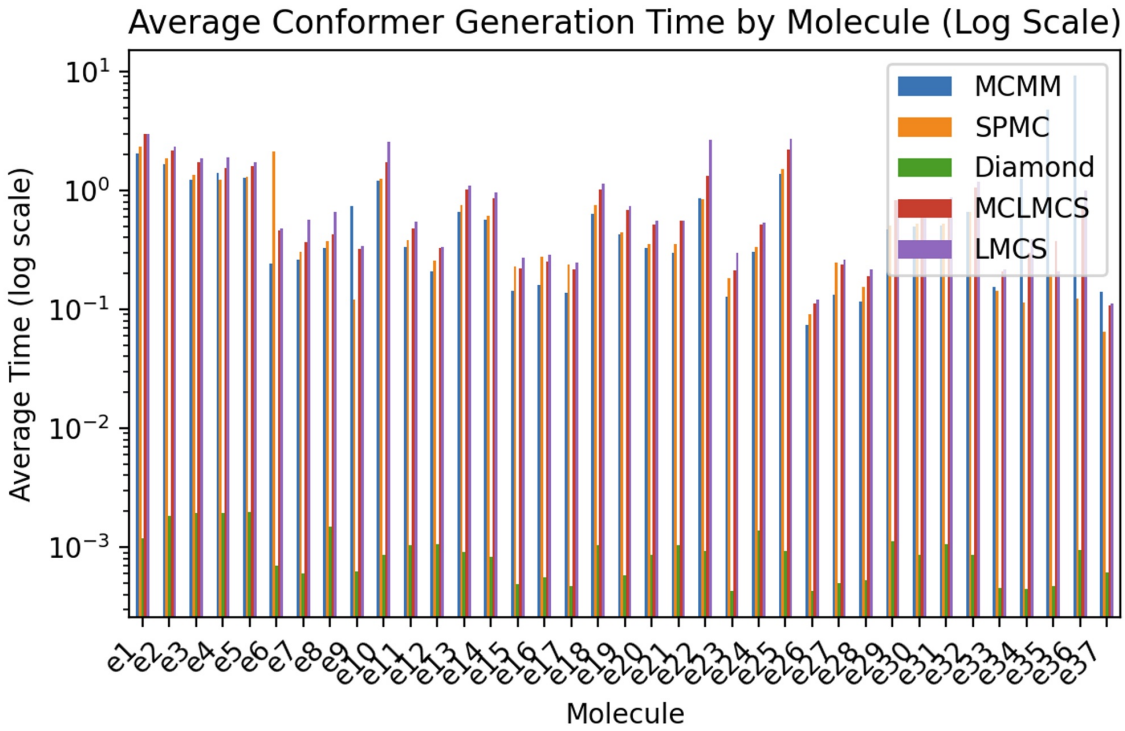


Figure 3.27 The average time cost to generate one conformer of various methods: *Diamond Energy* II, SPMC, MCMM, LMCS, MCLMCS.

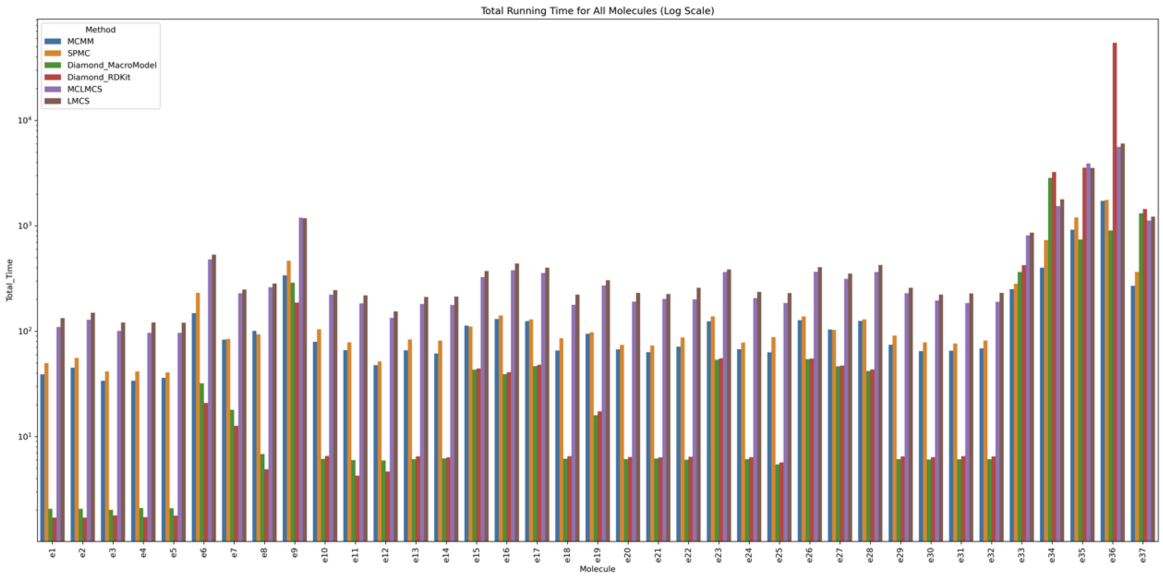


Figure 3.28 The total running time cost to do a complete conformer search for all the test molecule by various methods: *Diamond Energy* II two pipelines, SPMC, MCMM, LMCS, MCLMCS, using MM3.

Table 3.20 Total Running Time Cost of Each Method to Generate Conformers for All Test Cases Using MM3 Calculations.

Molecule	Diamond_MacroModel	Diamond_RDKit	MCMM	SPMC	MCLMCS	LMCS
e1	2.066	1.705	39.010	49.790	109.440	132.870
e2	2.066	1.706	44.960	55.800	128.480	149.450
e3	2.016	1.786	33.790	41.570	100.850	121.030
e4	2.096	1.716	33.820	41.510	96.600	120.780
e5	2.086	1.776	36.120	40.680	96.570	120.430
e6	31.986	20.805	148.210	230.910	479.920	532.610
e7	17.933	12.670	83.080	84.710	228.510	248.450
e8	6.837	4.897	100.790	93.170	261.280	283.380
e9	287.590	187.038	338.720	467.210	1192.770	1179.210
e10	6.142	6.554	79.380	104.070	222.390	245.640
e11	5.971	4.260	65.960	78.340	183.660	218.730
e12	5.942	4.657	47.530	51.510	133.900	154.380
e13	6.101	6.481	65.730	83.590	181.450	211.440
e14	6.211	6.343	61.550	81.280	177.570	212.860
e15	43.057	44.264	113.140	110.760	325.510	371.970
e16	39.157	40.814	130.600	141.110	377.920	439.670
e17	46.517	47.948	124.430	129.520	356.310	400.660
e18	6.172	6.512	65.530	85.580	178.090	222.540
e19	15.961	17.382	94.410	97.650	271.890	305.030
e20	6.111	6.412	67.170	74.100	190.820	230.760
e21	6.192	6.342	63.270	73.010	202.470	224.750
e22	6.032	6.442	71.350	87.170	200.300	257.440
e23	53.693	55.203	123.780	137.730	364.330	384.280
e24	6.123	6.391	67.360	78.050	206.040	236.050
e25	5.451	5.681	63.020	88.000	184.810	230.060
e26	54.340	55.103	127.420	137.780	366.460	405.100
e27	46.349	47.148	103.510	102.840	313.320	352.650
e28	41.854	43.287	124.900	129.450	364.520	424.570
e29	6.122	6.493	74.380	90.850	228.910	257.900
e30	6.051	6.391	64.740	78.290	195.990	221.820
e31	6.082	6.512	65.240	76.230	185.450	228.320
e32	6.112	6.491	68.490	81.380	190.200	230.690
e33	364.108	423.268	250.740	282.140	812.500	860.690
e34	2844.455	3230.976	399.790	731.550	1535.040	1785.740
e35	742.368	3546.624	916.600	1200.140	3904.810	3540.110
e36	903.386	54492.329	1721.360	1763.010	5597.030	6036.830
e37	1309.421	1443.357	268.740	365.360	1122.000	1222.070

Table 3.21 Average Running Time Cost of Each Method to Generate Conformers for All Test Cases Using MM3 Calculations.

Molecule	Diamond_MacroModel	Diamond_RDKit	MCMM	SPMC	MCLMCS	LMCS
e1	0.69	0.57	19.51	24.90	54.72	66.44
e2	0.69	0.57	14.99	18.60	42.83	49.82
e3	0.67	0.60	11.26	13.86	33.62	40.34
e4	0.70	0.57	11.27	13.84	32.20	40.26
e5	0.70	0.59	12.04	13.56	32.19	40.14
e6	0.80	0.52	4.49	7.45	20.00	22.19
e7	0.72	0.51	4.37	4.46	12.03	19.11
e8	0.76	0.54	6.30	5.82	16.33	25.76
e9	0.89	0.58	3.76	3.57	17.04	17.09
e10	0.68	0.73	19.85	26.02	55.60	81.88
e11	0.66	0.47	5.07	6.03	14.13	16.83
e12	0.66	0.52	2.97	3.22	8.37	9.65
e13	0.68	0.72	10.96	13.93	30.24	35.24
e14	0.69	0.71	8.79	11.61	25.37	30.41
e15	0.72	0.74	3.14	3.08	9.04	11.62
e16	0.73	0.76	3.63	3.92	10.50	12.56
e17	0.72	0.74	3.36	3.50	9.63	10.83
e18	0.69	0.72	10.92	14.26	29.68	37.09
e19	0.67	0.72	8.58	8.88	24.72	27.73
e20	0.68	0.71	5.17	5.70	14.68	17.75
e21	0.69	0.71	4.87	5.62	15.58	17.29
e22	0.67	0.72	14.27	17.43	40.06	85.81
e23	0.72	0.74	2.88	3.36	8.47	11.65
e24	0.68	0.71	4.81	5.58	14.72	16.86
e25	0.68	0.71	21.01	29.33	61.60	76.69
e26	0.73	0.74	1.47	1.66	4.21	4.66
e27	0.71	0.73	3.04	3.21	9.22	10.37
e28	0.72	0.75	2.60	2.75	7.59	8.85
e29	0.68	0.72	7.44	9.09	22.89	25.79
e30	0.67	0.71	7.19	8.70	21.78	24.65
e31	0.68	0.72	7.25	8.47	20.61	25.37
e32	0.68	0.72	9.78	11.63	27.17	32.96
e33	0.67	0.78	3.30	4.48	12.50	12.85
e34	0.83	0.94	7.01	6.84	21.93	30.79
e35	1.29	1.27	12.06	13.34	43.87	21.59
e36	37.64	1.45	18.31	7.60	91.76	137.20
e37	0.81	0.89	1.98	1.84	6.03	6.27

Table 3.22 Total Running Time Cost of Each Method to Generate Conformers for All Test Cases Using OPLS4 Calculations.

Molecule	Diamond_MacroModel	Diamond_RDKit	MCLMCS	MCMM	SPMC	LMCS
e1	23.68	19.28	83.79	30.77	40.93	91.64
e2	23.84	19.94	88.12	36.84	42.69	101.21
e3	23.38	19.72	63.85	27.94	32.47	76.15
e4	23.51	19.04	62.12	29.69	35.21	77.82
e5	23.39	18.70	61.30	28.39	36.87	77.96
e6	333.93	214.80	261.06	97.00	415.96	291.03
e7	182.88	129.06	136.83	54.87	61.16	154.72
e8	70.26	48.03	160.40	58.34	60.59	176.03
e9	3017.44	1925.17	713.81	223.56	222.58	731.15
e10	71.14	71.52	135.80	50.38	56.05	155.31
e11	70.13	64.19	111.01	39.68	46.62	130.65
e12	69.02	62.53	79.56	34.53	35.00	89.75
e13	69.85	70.23	107.07	40.62	47.07	124.21
e14	69.45	70.37	106.69	40.75	49.20	122.06
e15	475.40	486.77	186.79	71.04	70.33	208.70
e16	425.57	437.23	223.22	78.80	80.10	239.95
e17	511.62	527.97	196.73	71.76	73.60	222.19
e18	70.30	70.52	106.94	40.41	51.75	131.58
e19	185.53	189.06	161.75	64.68	59.76	181.24
e20	70.08	70.31	109.08	39.41	46.83	133.55
e21	70.21	70.36	111.02	43.55	44.36	129.71
e22	69.85	70.44	108.60	43.25	45.31	125.74
e23	573.20	587.37	206.16	71.43	68.83	229.15
e24	69.54	70.11	117.28	44.22	52.18	140.65
e25	62.19	62.52	137.31	51.67	57.56	161.16
e26	577.19	589.04	228.56	80.29	76.95	262.06
e27	499.25	512.24	208.69	71.89	74.23	248.60
e28	450.99	461.81	215.23	83.80	76.42	243.90
e29	70.61	70.63	144.61	50.91	55.98	176.63
e30	69.49	70.30	108.56	38.33	45.72	125.66
e31	69.56	70.41	102.29	38.05	45.57	133.83
e32	70.10	70.08	130.23	42.48	47.09	138.51
e33	3628.52	4296.22	479.39	148.99	141.06	516.46
e34	22693.16	30499.23	1189.80	326.51	354.66	1237.93
e35	5719.92	30637.59	2199.95	778.33	675.43	2369.16
e36	1309.83	411894.36	4014.02	1511.65	918.76	4114.43
e37	12606.08	14852.86	670.40	170.18	194.11	717.57

Table 3.23 Average Running Time Cost of Each Method to Generate Conformers for All Test Cases Using OPLS4 Calculations.

Molecule	Diamond_MacroModel	Diamond_RDKit	MCLMCS	MCMM	SPMC	LMCS
e1	7.89	6.43	27.93	15.39	20.47	30.55
e2	7.95	6.65	29.37	12.28	14.23	33.74
e3	7.79	6.57	31.93	13.97	16.24	38.08
e4	7.84	6.35	31.06	14.85	17.61	38.91
e5	7.80	6.23	30.65	14.20	18.44	38.98
e6	8.35	5.37	7.25	2.69	11.89	10.39
e7	7.32	5.16	7.20	2.89	3.22	8.14
e8	7.81	5.34	5.94	2.16	2.24	6.52
e9	9.37	5.98	7.51	1.56	2.20	7.70
e10	7.91	7.95	11.32	4.20	4.67	12.94
e11	7.79	7.13	5.84	2.09	2.45	6.88
e12	7.67	6.95	7.96	3.45	3.50	8.98
e13	7.76	7.80	17.85	6.77	7.85	41.40
e14	7.72	7.82	13.34	5.09	6.15	15.26
e15	7.92	8.11	4.92	1.87	1.90	5.49
e16	7.88	8.10	8.27	2.92	2.97	9.23
e17	7.87	8.12	4.58	1.71	1.75	5.17
e18	7.81	7.84	17.82	6.74	8.63	43.86
e19	7.73	7.88	5.99	2.40	2.21	6.71
e20	7.79	7.81	5.74	2.07	2.47	7.03
e21	7.80	7.82	5.84	2.29	2.34	6.83
e22	7.76	7.83	9.05	3.60	3.78	10.48
e23	7.65	7.83	3.33	1.15	1.41	3.70
e24	7.73	7.79	9.02	3.40	4.35	10.82
e25	7.77	7.82	15.26	5.74	6.40	17.91
e26	7.70	7.85	2.10	0.72	0.99	2.34
e27	7.68	7.88	4.09	1.41	1.46	4.88
e28	7.78	7.96	2.31	0.91	0.84	2.62
e29	7.85	7.85	12.05	4.24	4.67	14.72
e30	7.72	7.81	18.09	6.39	7.62	20.94
e31	7.73	7.82	17.05	6.34	7.60	22.31
e32	7.79	7.79	9.30	3.03	3.37	9.89
e33	6.67	7.90	3.69	1.14	1.74	3.94
e34	6.61	8.89	4.44	1.60	3.38	4.71
e35	9.93	11.01	5.66	3.64	6.89	6.46
e36	54.58	10.92	10.04	9.11	3.18	10.06
e37	7.75	9.13	2.68	1.03	1.96	2.77

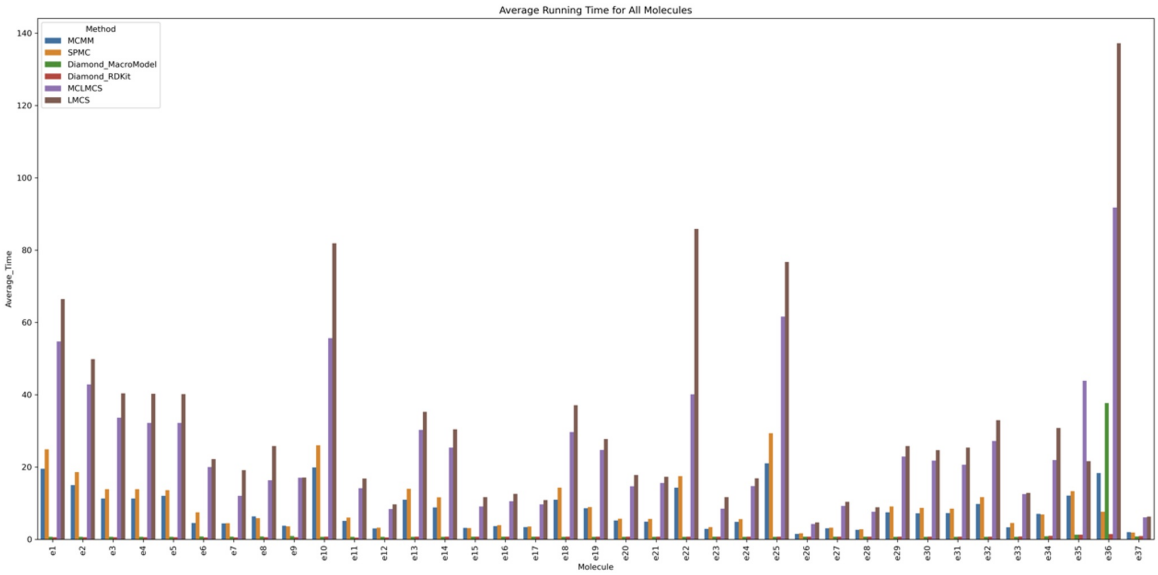


Figure 3.29 Average Running Time Cost of Each Method to Generate Conformers for All Test Cases Using MM3 Calculations.

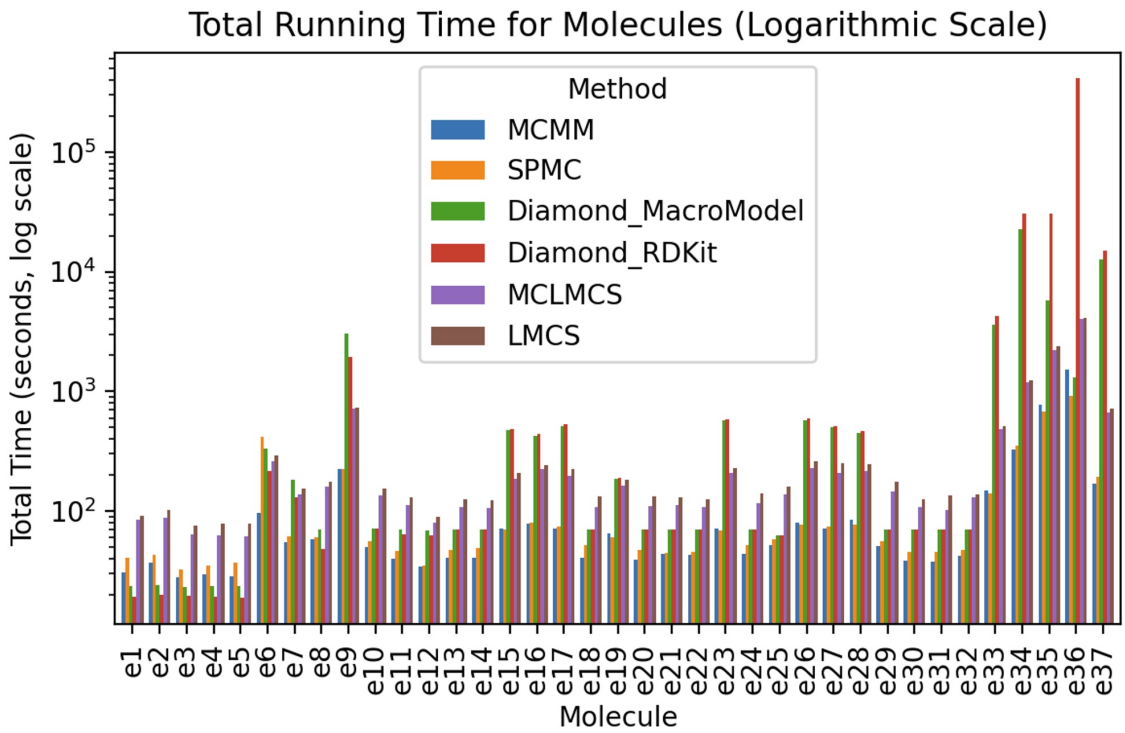


Figure 3.30 Total Running Time Cost (Logarithmic Scale) of Each Method to Generate Conformers for All Test Cases Using OPLS4 Calculations.

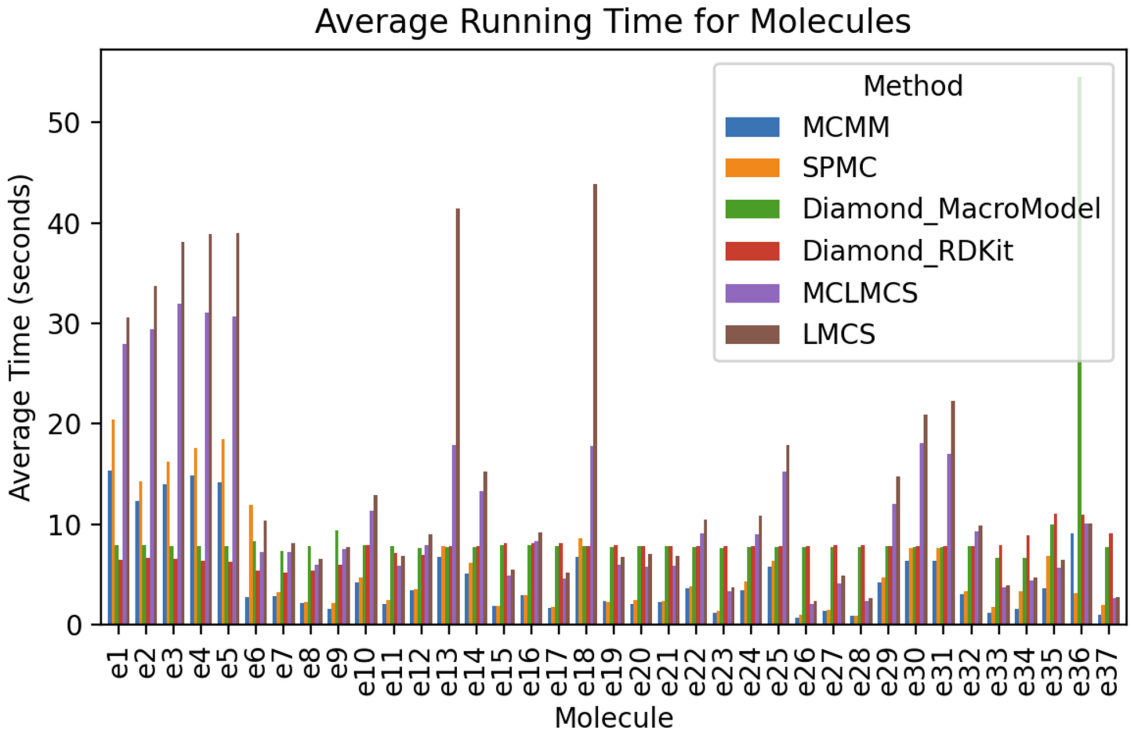


Figure 3.31 Average Running Time Cost (Seconds) of Each Method to Generate Conformers for All Test Cases Using OPLS4 Calculations.

LMCS (brown). The green and red bars are noticeably lower than the bars for other methods, often approaching the x-axis. These findings demonstrate that the *Diamond Energy* algorithm is faster in a full conformer search round compared to other methods.

The total time taken to complete the conformer search using the OPLS4 force field for the six pipelines: Diamond_MacroModel, Diamond_RDKit, SPMC, MCMM, MCLMCS, and LMCS. This data is detailed in Table 3.22, which covers the complete total running time, and Table 3.23, which provides the average total running time for each method.

Tables 3.22 and Table 3.23 provide molecular names in the first column and show the total and average running times for the Diamond_MacroModel and Diamond_RDKit pipelines in the subsequent columns. Comparative running times for the other methods, MCMM, SPMC, MCLMCS, and LMCS, are listed in the remaining columns. In OPLS4 force field calculations, MCMM and SPMC generally finish the full conformer search faster than most test cases, followed by Diamond_MacroModel, Diamond_RDKit, MCLMCS, and LMCS. Figures 3.30 and Figure 3.31 graphically depict these times, with a logarithmic scale for total running times and a linear scale for average times on the y-axis, and sample molecules on the x-axis. Diamond_MacroModel and Diamond_RDKit are shown in green and red, while MCMM, SPMC, MCLMCS, and LMCS are in blue, yellow, purple, and brown, respectively. The data suggests that the *Diamond Energy* II algorithm may be more aligned with MM3 force field evaluations rather than OPLS4, leading to longer minimization times with OPLS4 compared to MCMM and SPMC. This indicates that despite *Diamond Energy* II's speed in generating geometries, its overall performance in the minimization step is not as markedly fast as other methods using OPLS4 calculation.

Conclusions The data on geometry generation and full conformer search times using MM3 and OPLS4 indicate that *Diamond Energy* II is among the faster methods, particularly in generating geometries for complex sugar-like molecules. It outperforms other geometry generation strategies by several orders of magnitude. This performance is more pronounced in complex systems compared to the simpler molecules like alkanes used in the first version of the algorithm. These results affirm *Diamond Energy* II's effectiveness in dealing with realistic and complex molecular systems.

3.4 Conclusions

Diamond Energy II processes saccharides by organizing them into configurations that echo the geometry of diamond structures. This process validates the underlying hypothesis that diamond-like lattice structures serve as a heuristic for rapidly identifying low-energy configurations. Empirical testing confirms the algorithm's efficacy in pinpointing local energy minima informed by diamond-like structures, thereby improving the prediction of stable molecular configurations.

In molecular conformational analysis using MM3 and OPLS4, *Diamond Energy* II demonstrates superior performance. It exceeds the computational speeds of four established methods—SPMC, MCMM, LMCS, and MCLMCS—by orders of magnitude, particularly in complex saccharide analysis. In terms of locating minimum energy conformations, *Diamond Energy* II matches or exceeds these methods, notably in discovering a diverse set of stable conformers. This is evidenced by the number of conformers *Diamond Energy* II identifies within 3 and 5 kcal/mol range from the lowest energy state, illustrating its search accuracy.

However, limitations are observed. Utilizing the OPLS4 method, inclusive of a minimization procedure, may increase computation time while maintaining effectiveness. The algorithm faces computational burdens when assessing molecules with more than seventeen rotatable bonds due to the proliferation of potential configurations. It also shows limitations with saccharides comprising more than four six-membered rings, which negatively influences accuracy and prolongs the search process.

Future improvements will aim to address these issues.

Chapter 4

Diamond Energy Extension Work – Diamond AI

As the availability of high-quality QM calculation datasets—such as QM9, [128] GEOM, [128] and ISO17 [108]—continues to increase, alongside the development of increasingly robust machine learning models, new efforts are emerging to understand the patterns underlying low-energy conformers and to predict them more accurately. Given the inherent complexity and the vast dimensionality of a molecule's conformational space, his presents a significant computational challenge.

The following section will explore the potential impact and role of machine learning methodologies in expediting the search for low-energy conformations, approaching a *Diamond Energy* strategy.

4.1 Leveraging *Diamond Energy* Data for the Accelerated Identification of Low-Energy Conformers Through Machine Learning

4.1.1 Background

Machine learning has been extensively employed in recent years for molecular generation, particularly for the generation of compounds with specific properties. Typically, these models are trained on large, high-quality chemical databases such as ZINC [129] and ChEMBL [130]. However, there are fewer studies focusing on using machine learning to generate molecular conformations.

Early efforts in predicting molecular conformations predominantly focus on protein structure prediction. A classical work is AlphaFold [131], which predicts the most probable 3D structure of a given protein sequence.

The data used by AlphaFold encompasses a mix of publicly available datasets of experimentally determined protein structures, such as the Protein Data Bank (PDB) [92]. Additionally, it taps into large-scale sequence databases like UniRef [132] to get evolutionary insights. Through the analysis of multiple sequence alignments, AlphaFold recognizes conserved amino acid positions across related proteins. It further detects covariation patterns, where two amino acid positions evolve concurrently. Such covarying pairs often signify physical contact or a functional relationship within the protein's 3D structure.

The extraction of this information from data is accomplished using a transformer-based model, equipped with a self-attention mechanism. Transformers are potent tools for natural language processing (NLP) tasks. They excel in weighing the significance of various segments of an input sequence when generating an output. This capability allows the model to consider a sequence's context, capturing both short and long-range dependencies without relying on recurrent structures. This is achieved primarily through the self-attention mechanism.

Imagine a person reading a sentence and emphasizing or "attending to" specific words that offer context or clarity for comprehending another word in the sentence. When faced with a complex sentence, individuals might intuitively link certain words to grasp their deeper meaning. Similarly, the self-attention mechanism computationally determines which segments of the input sequence are important for understanding its entirety.

Additionally, transformers incorporate multi-head attention, which permits the model to attend to various portions of the input for distinct reasons, all at the same time. This design provides transformers with the unique capability to process entire sequences simultaneously. Unlike human beings, who tend to process information step by step when reading, transformers are powerful and can process entire sequences simultaneously. Although they don't "understand" text in the same way humans do, they do however recognize patterns based on their training data. Such descriptions are mere simplifications; the underlying mathematical and computational intricacies of these models are far more nuanced than these illustrative analogies.

In AlphaFold, the transformer evaluates the relevance of each amino acid residue within the entire sequence's context. This mechanism enables the model to identify distant interactions between amino acid residues, which are crucial for grasping protein folding. For every pair of amino acids in the protein, the model estimates their mutual distance, establishing a spatial interrelationship map. This map subsequently aids in predicting the protein's 3D structure. The output from a standard AlphaFold run is a single predicted structure. However, in reality, proteins can adopt multiple conformations depending on their environment, interactions with other molecules, or due to inherent flexibility.

Predicting the potential conformational distribution of a given molecule presents a challenging task. The inherent complexity arises from the fact that a single molecule can adopt multiple 3D conformations, making this a multi-modal learning task. Despite the challenge, some attempts have been made.

The earliest machine learning work reported to do small molecule conformations prediction is GraphDG [107], which is a probabilistic model designed to generate conformers from molecular graph representations. They utilized the CONF17 dataset, which they derived from ISO17 by extracting and refining molecular conformer data. After the refinement process, the dataset contains 197 molecules, all of which are isomers of $C_7H_{10}O_2$, with a total of 430,692 conformers. ISO17 is based on snapshots from ab initio molecular dynamics simulations of these isomers from the QM9 dataset. All their conformers are represented as undirected molecular graphs and also a set of atomic distances. Just considering the primary bonds (or connections) between atoms is not enough to capture the full 3D shape. Considering this, they also add extra connections to their molecular graph. Some of these connections help to understand the angles between atoms, especially for atoms that are second neighbors in original graph G. They also add connections that help to determine dihedral angles, and the method is grounded in distance geometry effectively accounting

for the rotation and translation invariance of molecular conformations, bringing potentially promising performance.

GraphDG model is based on a Conditional Variational Autoencoder (CVAE). Autoencoder (AE) is a neural network architecture that aims to learn efficient representations of input data. It consists of two main parts: an encoder that compresses the input into a compact latent representation, and a decoder that reconstructs the input from this representation. It is primarily designed for dimensionality reduction and feature learning. Autoencoder (AE) is not designed for generation purposes, its primary aim is to reproduce the input data as accurately as possible in the output. Variational Autoencoder (VAE) is built and extended on AE as a generative model. In VAE, the encoder does not map an input directly to a fixed vector in the latent space. Instead, it maps the input to a distribution (typically Gaussian) described by mean and variance. During training, a sample is drawn from this distribution and is fed into the decoder. The latent space of a VAE is more densely populated. If you sample a random point from the VAE latent space and decode it, you will be likely to get a meaningful output similar to the training data. Conditional Variational Autoencoder (CVAE) is then built on VAE, and it aims to learn a latent representation of the input data conditioned on some additional information to generate new data that not only is similar to the input but also conforms to the provided condition. In their work, molecular graph G is being used as the conditioning variable. The input for their CVAE model is a molecular graph G and distances d, and the final output is a set of distances d. So, given a molecular graph G, the CVAE models the distribution of possible distances d (or molecular conformations) that are consistent with that graph. This allows the model to generate multiple potential distance matrices (or conformers) for a given molecular graph, reflecting the inherent flexibility and variability of molecular structures.

While it is noted that GraphDG is not simply a CVAE model, and the center of their model architecture is message-passing neural networks (MPNNs). Imagine the points of the molecular graph as people in a network where they talk (or exchange "messages") with their neighbors to learn more about the entire network. Similarly, the connections between these points (called edges) also update based on the points they connect. MPNNs are particularly good at understanding and representing molecules.

Subsequently, a model termed Conditional Graph Continuous Flow (CGCF) [109] was introduced. Acknowledging the sensitivity of Cartesian coordinates to changes in rotation and translation, the CGCF authors, similar to the approach of GraphDG, chose to ground their generative model in inter-atomic distances, enhanced by auxiliary bonds. In contrast to

GraphDG, which translates atomic pairwise distances into 3D geometry using distance geometry techniques, CGCF articulates conformations using a conditional distribution formula.

Within the CGCF framework, a Conditional Variational Autoencoder (CVAE) augmented with MPNNs is employed. What sets it apart is the encoder's handling of the latent space. Rather than relying on a basic Gaussian distribution for the CVAE encoder's latent space, CGCF incorporates Continuous Normalizing Flows (CNFs). This approach starts with a standard Gaussian distribution and then evolves it into a more intricate distribution using CNFs, enabling the model to better capture and represent the authentic data distribution.

Following this encoding phase, CGCF uses probabilistic modeling to derive conformations from distance matrices. Additionally, the framework integrates an Energy-based Tilting Model (ETM) towards its conclusion. ETM refines the generated conformations, ensuring their energetic feasibility. It is worth noting that the underpinning of the ETM is SchNet [108], an architecture known for modeling molecular energy profiles.

In their research, the CGCF authors employed a dataset that was more comprehensive than what was used in GraphDG. Their data collection not only includes the ISO17 dataset but also incorporates two recent datasets: GEOM-Drugs and GEOM-QM9. GEOM-QM9 serves as an extension of the widely recognized QM9 dataset. While the original QM9 offers a single conformation for each molecule, GEOM-QM9 presents multiple conformations. This dataset primarily centers on smaller molecules, characterized by a cap of 9 heavy atoms (totaling 29 atoms), modest molecular mass, and a restricted count of rotatable bonds. In contrast, GEOM-Drugs is a substantially larger dataset compared to GEOM-QM9. It features up to 181 atoms, with 91 of them being heavy atoms. On average, the molecules in this dataset possess 6.5 rotatable bonds. Like GEOM-QM9, GEOM-Drugs also provides multiple conformations for each molecule.

A subsequent advancement in this field was the development of ConfVAE [111], designed to improve upon the two-stage approach used by GraphDG. While GraphDG begins by predicting inter-atomic distances and then optimizes these to form a 3D structure using distance geometry, ConfVAE adopts an end-to-end method. It is based on a conditional variational autoencoder (VAE) framework, much like GraphDG, and also integrates message-passing neural networks (MPNNs) at its core to process molecular data. The key difference between ConfVAE and GraphDG is in their application: instead of employing the CVAE framework to initially determine inter-atomic distances for multiple conformers and then

converting them into a 3D geometry using a distance matrix, ConfVAE directly outputs the 3D structures through a bilevel optimization program.

Bilevel optimization is characterized as a method where the variables in the primary (outer) objective function are determined by resolving a secondary (inner) optimization problem. This approach has demonstrated efficacy in multi-task learning. Within the ConfVAE structure, the outer objective is minimizing the generalization error on 3D structures, ensuring that the generated conformation aligns with the ground truth, considering factors like rotation and translation. The inner objective, on the other hand, focuses on bridging the gap between the predicted and actual interatomic distances. To highlight the significance of this bilevel optimization in ConfVAE work, they conducted an ablation study. The results revealed a performance decline upon removal of the bilevel component. This outcome supports the idea that exclusively deriving knowledge from distances can introduce biases in the generated conformations. An end-to-end approach, which emphasizes direct learning from the 3D structure effectively addresses and mitigates this issue. The dataset used in ConfVAE are the same as that of CGCF.

Recently, the GeoMol [110] model emerged as a novel approach to molecular structure prediction. Distinct from prior methods, which primarily learned atom pair distance distributions within a CVAE framework, GeoMol adopts an end-to-end graph-to-graph strategy. This model employs Message Passing Neural Networks (MPNNs) to effectively capture both local and global molecular interactions.

GeoMol's primary objective is to predict local atomic 3D configurations, which include bond distances, bond angles, and chiral centers, as well as torsion angles. A standout feature of GeoMol is its unique loss function designed to promote the generation of diverse, low-energy conformers. This loss function contains:

I. Single Conformer Loss Instead of comparing atom positions directly in 3D space, GeoMol measures differences in the distributions of predicted geometric properties (like distances and angles) against known structures. Specifically, it employs Normal Distributions for distance evaluations and von Mises Distributions for angle evaluations. This method, as believed by the authors, is more efficient than the conventional Root Mean Square Deviation (RMSD) approach. II. Node Symmetry Loss In molecular graphs, certain atoms can appear symmetric, posing challenges in differentiation. Though GeoMol initially attempts distinction using random noise vectors, it occasionally misaligns with true 3D positions. To address this, the authors introduced a loss function that evaluates all potential arrangements of symmetric

nodes to best align with the actual 3D configuration. III. Ensemble Conformer Loss Given the uncertainty around the exact number of true conformers and their direct correspondence with generated conformers, GeoMol employs an Optimal Transport (OT) based approach. OT establishes a transport plan, a matrix detailing the transformation of each element in one distribution to match another. In essence, the OT-based approach offers a sophisticated way to compare two distributions, especially when the exact correspondence between their elements is unclear. There's a cost associated with each of these transformations, which can be thought of as the "distance" between the elements being matched. The goal is to find a transport plan that minimizes the total cost. In other words, it finds the most efficient way to make the generated conformers match the true conformers.

GeoMol's sophisticated loss function integrates three distinct terms, each representing different prediction costs. During training, the model's objective is to minimize this composite loss, ensuring precise and diverse molecular structure predictions.

While many models above are designed to directly learn the distribution of conformations from the training set, there exists a novel strategy that employs a score-based model. Here, the term "score" does not imply a grade or rating, as it might in common sense. Instead, it signifies a mathematical concept known as the "score function." This function measures how a data point's likelihood changes with slight attribute adjustments. Imagine visualize data points on a landscape; the score function would reveal the gradient at every spot, providing insights into the direction and magnitude of changes in the data's probability distribution. Inspired by traditional force field methods used in molecular dynamics simulations, an innovative approach named ConfGF [112] has been developed. It focuses on directly estimating the gradient fields of the log density of atomic coordinates through a score-based model.

ConfGF is designed to generate molecular conformations by leveraging the "score" or gradient fields of atomic coordinates. This gradient field effectively shows the direction and degree of change in the atomic coordinates' probability distribution. Using basic Graph Neural Networks (GNNs) to directly capture this gradient field is challenging because these networks struggle to handle rotations and translations. As a solution, the focus shifts to interatomic distances rather than the atoms' Cartesian coordinates. This approach is favored because interatomic distances offer a consistent representation of a molecule, irrespective of rotation or translation. Since these distances can be differentiated in terms of the Cartesian coordinates, the chain rule is employed to connect the gradients of both. After training, the model is not used for conformer samples directly. Instead, it employs Langevin dynamics,

a stochastic method, for this purpose. Guided by the trained score function, Langevin dynamics, commonly used in physics to describe particle motion in a fluid, is adapted to these score-based generative models. This adaptation allows it to sample new data points by using the function's gradient to push the sample toward higher likelihood regions, coupled with a random noise component to ensure diverse exploration. Like earlier models, ConfGF utilized the GEOM-drugs, GEOM-qm9, and ISO17 datasets.

Another model, named GeoDiff, [106] employs the diffusion model to identify the stable state of a molecule. This diffusion process draws inspiration from thermodynamics, where a system in equilibrium gradually transitions toward a more chaotic or "noisy" state over time. The authors draw an analogy between molecular conformations and thermodynamic systems, positing that stable molecular conformations can be predicted by reversing this diffusion process. Using a molecular graph as input for the trained model, its conformation is generated by first sourcing from the noisy state distribution, and then undergoes iterative refinement through the reverse Markov process to approach the equilibrium state.

In the field of machine learning for conformer generation, the quality of the training data often limits the performance of the models, making them dependent on underlying theories such as QM or MM. However, prior research has shown that the *Diamond Energy* method is not limited by theoretical constraints and can quickly identify low-energy molecular conformers. Given the extensive amount of experimental data available, using *Diamond Energy* experiments output geometric distributions as a training dataset to build a conformer prediction model is promising for speeding up conformational searches. To test this idea, data from *Diamond Energy* outputs have been collected and prepared as training sets for four specific machine-learning models. These models, once trained, were used to generate low-energy conformers. The generated results were then minimized and compared with those produced by the original *Diamond Energy* method to evaluate their effectiveness.

Currently, four models can accept a molecule as input and subsequently output multiple low-energy conformers by learning patterns from these conformers during training: GraphDG, CGCF, ConfVAE, and GeoMol. Our work plan involves training these models using data from conformers that closely align with the diamond lattice framework, and the objective is to evaluate the models' ability to quickly generate diamond-like conformers.

In previous tests, it was observed that the minimization step significantly affected the speed at which final conformer search results were obtained. Thus, two models, ConfGF and Geodiff, which are assumed to be able to quickly obtain stable 3D geometry, are being compared with

the traditional energy minimization method, PRCG, to evaluate their minimization abilities in a complete conformer search.

4.1.2 Results and Discussion

4.1.2.1 Experimental Workflow

The experiment plan of this chapter is outlined below Figure 4.1. The study will utilize data from previous *Diamond Energy* projects and feed this data into leading machine learning models, which aim to learn the distribution of low-energy spaces. The goal is to determine whether structures resembling the low-energy lattice of *Diamond Energy* can guide these models to perform conformer searches more efficiently, without the need for quantum mechanics (QM) level data as a training set. Furthermore, a previous study found that minimization is the slow step in the full cycle search of *Diamond Energy*. Thus, models designed to find stable structures will be tested and compared with traditional minimization processes. This comparison aims to assess whether these machine learning-based models can improve the minimization step in real-world tasks.

4.1.2.2 Dataset

The dataset, derived from *Diamond Energy*'s results, has been compiled and processed to train statistical models on the geometry of conformers that resemble a diamond lattice. This training aims to assess how effectively these data-driven models can replicate geometric features and whether they can be employed for conformer generation.

To minimize bias in the results, each subset contains structurally similar molecules without any repetitions across the sets. Despite *Diamond Energy*'s systematic search capabilities, it encounters limitations when dealing with molecules having more than 18 rotatable bonds. To evaluate whether the statistical models can address this challenge, the test set includes new molecules that were not part of previous assessments, providing insights into the model's performance on previously unseen molecules.

This study utilizes a dataset comprising 78 unique molecular structures in Figure 4.2, Figure 4.3, Figure 4.4, and Figure 4.5.

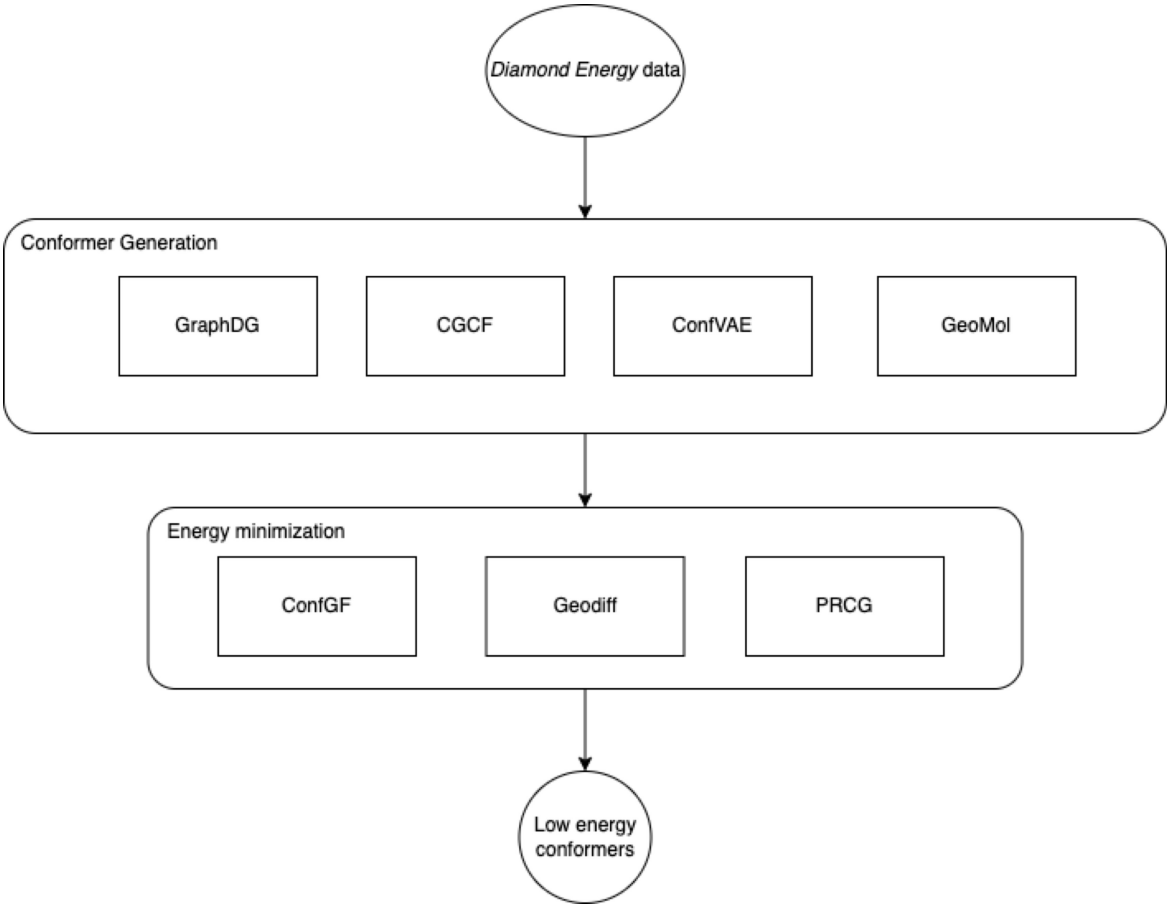


Figure 4.1 Workflow chart illustrating the use of *Diamond Energy* output data to train the current conformer search machine learning model.

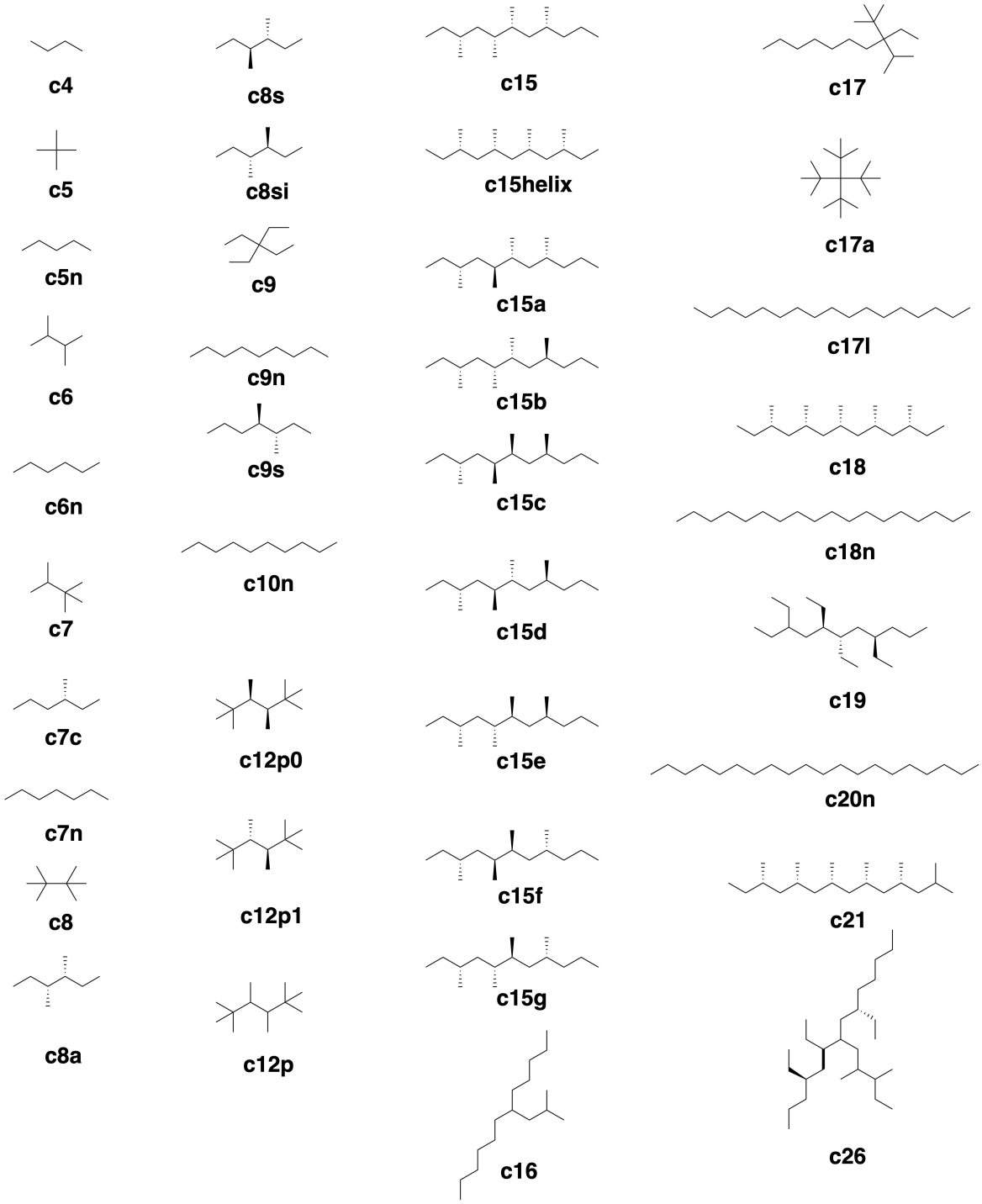


Figure 4.2 *Diamond Energy* dataset used in this study.

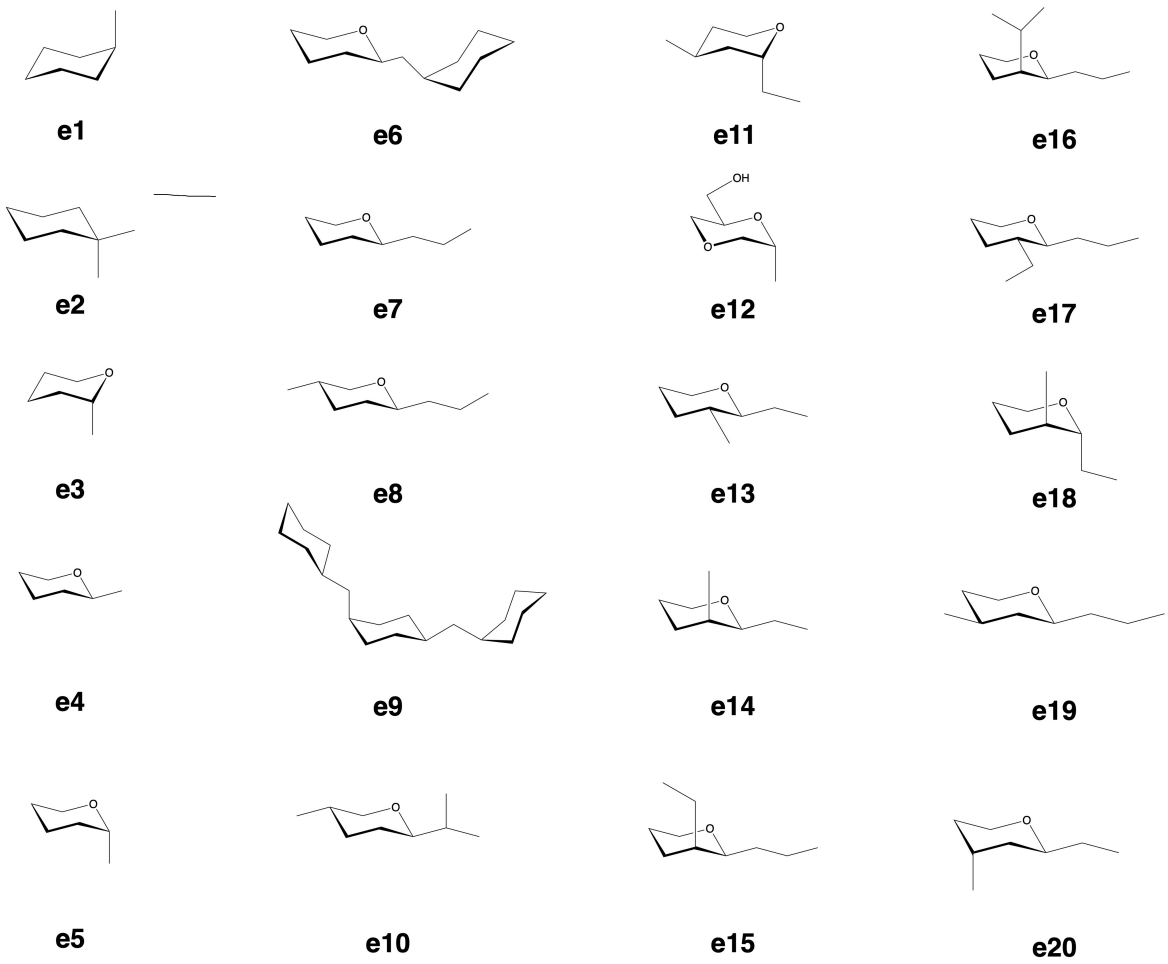


Figure 4.3 *Diamond Energy* dataset used in this study.

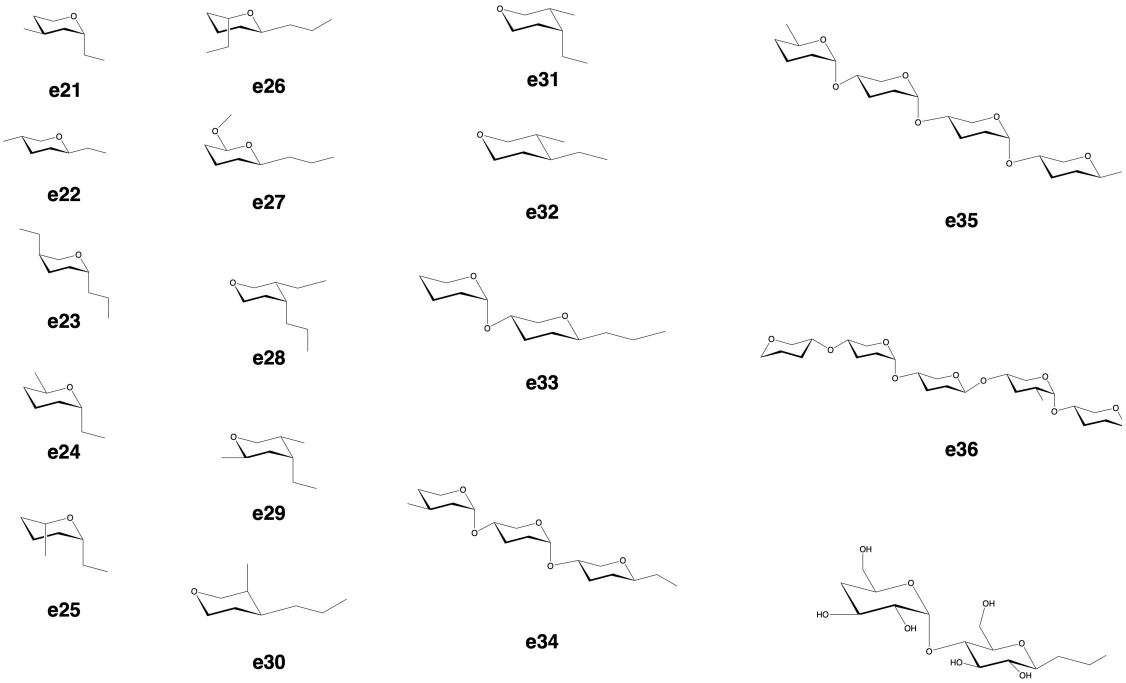


Figure 4.4 *Diamond Energy* dataset used in this study.

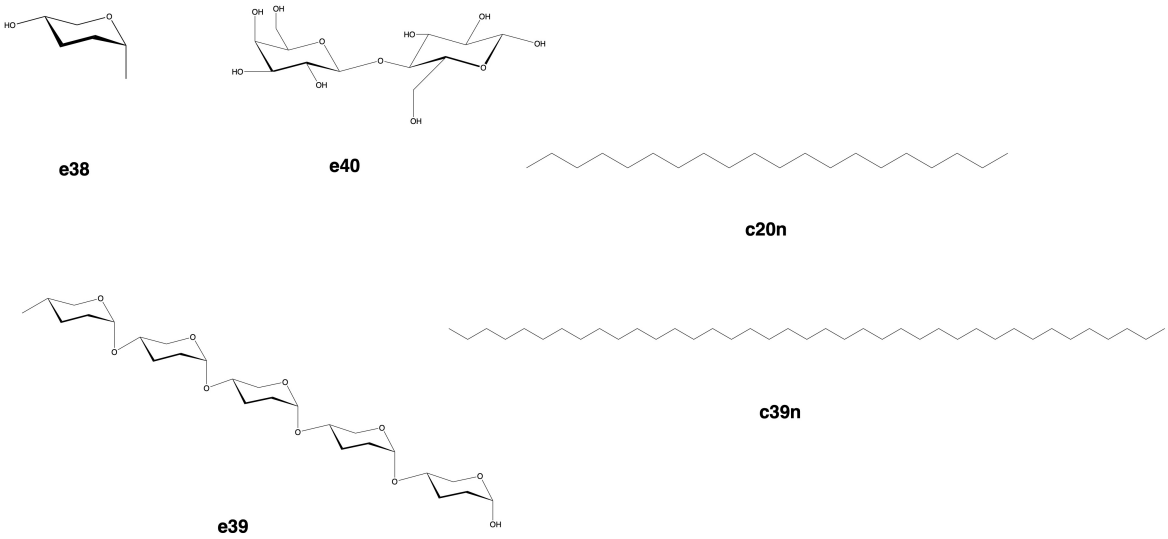


Figure 4.5 *Diamond Energy* dataset used in this study.

The dataset is divided into three parts: 57 molecules are used for the training set, 8 for the validation set, and 13 for the testing set, following an approximate ratio of 80:10:10. Each molecule's distinct conformer structures were obtained from the *Diamond Energy* program's results. The training set includes 68,158 conformers, the validation set contains 9,510 conformers, and the test set comprises 8,203 conformers.

The molecules are designated by various names, such as **c4**, **c5**, **c5n**, and so on, up to **e40**. The sets are carefully curated to ensure a diverse representation of molecular structures, with the training set including molecules like **c5**, **c6n**, **c8si**, among others. The validation set features molecules such as **c7c** and **e18**, while the test set includes **c4**, **c7n**, **c8a**, and additional molecules **c20n**, **c39n**, **e38**, **e39**, and **e40** not previously used in the training or validation phases.

Here is a breakdown of the specific molecules included in each set:

The training set is composed of molecules **c5**, **c5n**, **c6**, **c6n**, **c7**, **c8**, **c8s**, **c8si**, **c9**, **c9n**, **c9s**, **c10n**, **c12p0**, **c12p1**, **c15**, **c15a**, **c15b**, **c15c**, **c15e**, **c15f**, **c15g**, **c16**, **c17**, **c18**, **c19**, **c21**, **c26**, **e1**, **e2**, **e3**, **e4**, **e5**, **e6**, **e8**, **e10**, **e11**, **e13**, **e14**, **e15**, **e16**, **e17**, **e19**, **e21**, **e22**, **e23**, **e24**, **e25**, **e26**, **e28**, **e29**, **e30**, **e32**, **e33**, **e34**, **e35**, **e36**, **e37**. The validation set includes **c7c**, **c15d**, **c18n**, **e9**, **e7**, **e18**, **e20**, **e31**. The test set contains **c4**, **c7n**, **c8a**, **c12p**, **c15helix**, **c17l**, **c20n**, **c39n**, **e12**, **e27**, **e38**, **e39**, **e40**.

This composition ensures that similar structures are present in each set to maintain consistency, yet without any overlap of identical molecules across the sets. This approach tests the capability of statistical models to generalize and predict conformer structures for molecules with a high number of rotatable bonds, which is a known limitation in systematic search methods like *Diamond Energy*. The test set could evaluate the models' performance on both similar molecules as in the training set and new, unseen molecules.

4.1.2.3 Results and Discussion

Assessing the Performance of Conformer Generation Models: CGCF, GeoMol, and ConfVAE in Learning Diamond Lattice Low Energy Conformer Distributions To assess how well these three geometry generation models—ConfVAE, CGCF, and GeoMol—learn from *Diamond Energy*'s data, this study compares the geometric distribution of molecular conformers from the models to a benchmark dataset, namely *Diamond Energy*'s output of 3D conformers. Simple dihedral angle distributions are chosen for this comparison. Dihedral

angles are effective in capturing the three-dimensional shape of a molecule, and their distributions provide insights into the conformational flexibility and preferred conformations of the molecules. While other criteria, such as the radius of gyration, which reflects the molecule's compactness, and the distribution of interatomic distances, could be suitable choices, the moment of inertia tensor calculation for each conformer, which indicates the distribution of mass relative to different rotational axes, is also informative. However, given that the *Diamond Energy* data primarily comprise simple alkanes and common saccharides, the critical message currently pertains to the diamond lattice framework. Therefore, this study focuses solely on comparing the dihedral angle distributions of the model results with the actual *Diamond Energy* output.

The calculation is carried out as follows: first, we identified the indices of atoms that form dihedral angles and then read the SDF files for the test molecules. The bins in the histogram were normalized to create a probability density. The Wasserstein distance between these distributions is calculated to measure the amount of work required to make one distribution identical to another. According to the data presented in Table 4.1, the values appear to be very small. This indicates that the three models have generated dihedral angles that are quite similar to *Diamond Energy* torsion angles pattern.

Table 4.1 illustrating the gap between the test molecules' dihedral angle distributions for the three models and the benchmark *Diamond Energy* data's dihedral angle distributions. It seems that the methods are very close to the actual *Diamond Energy* data dihedral angle distributions. It should be noted that the GeoMol data lack ring-containing molecules, as these are outside the intended usage range of the GeoMol model.

Table 4.1 Comparison of Dihedral Angle Distribution Distances Between Various Models and *Diamond Energy* Data.

Molecule	**CGCF_closeness**	**ConfVAE_closeness**	**GeoMol_closeness**
c8a	0.003	0.003	0.003
c4	0.002	0.001	0.002
c17l	0.005	0.004	0.004
c12p	0.004	0.004	0.004
c15helix	0.004	0.004	0.004
c7n	0.003	0.003	0.003
e12	0.004	0.004	—
e27	0.004	0.004	—

To further quantify and analyze the data mentioned above, statistical calculations were performed on the dihedral angle distribution distances. As shown in Table 4.2, all the statistical measures are very similar across all three models. These measures include the Average Distance, Median Distance, Standard Deviation, Maximum Distance, and Minimum Distance. A Kruskal-Wallis test was also conducted to check the difference between these three models, with Kruskal-Wallis Test Statistic is 0.21, which means no big disparity among the groups, and P-value is 0.90, which means there is no statistically significant difference in the median distances across the three models.

Table 4.2 Statistic analysis of CGCF, GeoMol and ConfVAE closeness results.

Statistic	**CGCF_closeness**	**GeoMol_closeness**	**ConfVAE_closeness**
Average Distance	0.004	0.003	0.004
Median Distance	0.004	0.004	0.004
Standard Deviation	0.001	0.001	0.001
Max Distance	0.005	0.004	0.004
Min Distance	0.002	0.002	0.001

Evaluating Model Performance: The Impact of Training Data on ML-Based Models and Their Subsequent Comparison with the *Diamond Energy* Algorithm in Real-World Conformational Search Tasks Below is the work plan for this experiment (Figure 4.6). Different training datasets are utilized in their respective conformer search generation models. It will assess whether the output from *Diamond Energy* can effectively guide the model to better learn the distribution of low-energy conformers, thereby eliminating the need for costly quantum mechanical (QM) calculations during the preparation of the conformer search dataset. These machine learning based model's performance will also be compared with the *Diamond Energy* algorithm as conformational search tasks.

Variations in Training Data (Diamond, ISO17, QM9, DRUGs) and Their Impact on the Output of CGCF model The Table 4.3 presents the results of conformers generated by CGCF when trained on various datasets and subsequently energy-minimized using the RDKit MMFF94 force field. The first column lists the molecular names, while the subsequent columns display results corresponding to different training datasets: Diamond Energy data, ISO17, QM9, and DRUGs.

An ANOVA was then conducted to analyze the differences in the lowest energy values identified for all these pipelines using RDKit's MMFF94 force field calculations. The results

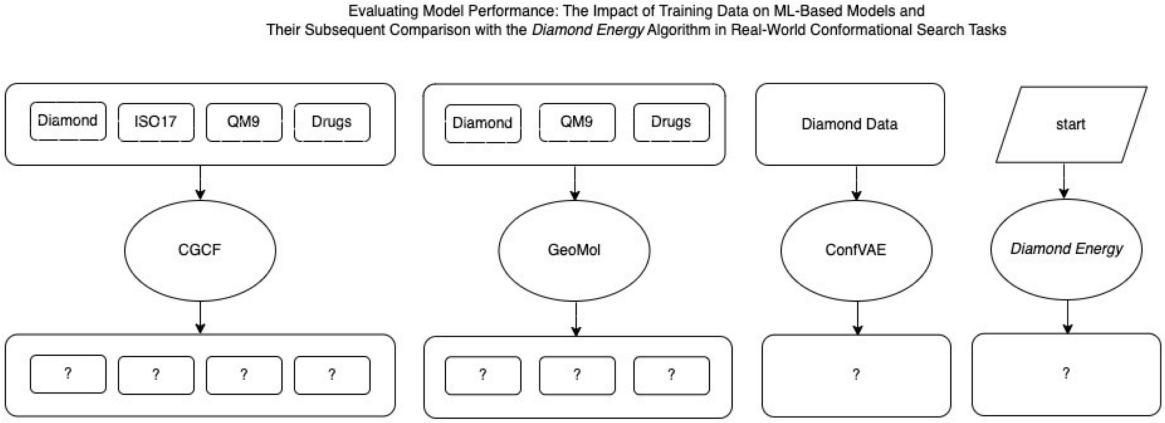


Figure 4.6 Flow Chart Comparing the Effects of Different Datasets on Conformer Search Generation Models and Subsequent Comparison with *Diamond Energy*.

Table 4.3 Comparative Results of CGCF Trained on Different Datasets Using MMFF94 Calculations (kcal/mol).

Molecule	CGCF_Diamond_MMFF94	CGCF_ISO17_MMFF94	CGCF_qm9_MMFF94	CGCF_drugs_MMFF94
c12p	47.28	47.28	47.28	47.28
c15helix	20.29	20.13	22.77	22.97
c17l	−4.39	−4.42	−2.96	−2.62
c20n	−5.02	−3.96	−3.82	−1.68
c39n	−8.57	−6.36	−6.54	−5.66
c4	−5.08	−5.08	−5.08	−4.29
c7n	−4.85	−4.85	−4.25	−4.05
c8a	8.37	8.37	8.37	8.18
e12	71.56	71.56	73.82	74.15
e27	−3.76	−3.38	−3.76	−2.23
e38	28.25	26.05	26.05	28.36
e39	35.48	38.61	32.54	44.60
e40	173.51	174.70	173.45	170.27

indicated an F-value of 0.003 and a P-value of 0.999, suggesting that there is no statistically significant difference between the groups. This implies that for the term 'lowest energy identified' for these test molecules, the training data did not have a significant impact on the result. At least for this outcome, it does not support the notion that the *Diamond Energy* data could significantly improve the CGCF model's performance in finding the lowest energy value using the MMFF94 force field.

Similar calculations were also conducted using the MM3 and OPLS4 force field calculations with MacroModel. Data from previous experiments involving *Diamond Energy* calculations are available, so the comparison henceforth will incorporate the *Diamond Energy* conformer search process as a benchmark for real-world conformational search task performance.

The following table, Table 4.4, presents the results of CGCF trained on different datasets and the systematic search algorithm *Diamond Energy* (RDKit pipeline) using MM3 calculations. Some molecules, such as **c20n**, **c39n**, **e38**, **e39**, and **e40**, were previously out of the range of the *Diamond Energy* algorithm. Consequently, data are missing in the final column, which represents the results of *Diamond Energy*. It can be observed that, although *Diamond Energy* identifies the lowest energy value in most test cases, the gap between the groups is not very large. Similarly, an ANOVA test was performed for each pipeline to determine if significant differences exist between the groups. The results indicated an F-value of 0.016 and a P-value of 0.999, which means that there is no significant difference between all five pipelines. In other words, the results of CGCF in identifying the lowest energy value for test molecules are acceptable, and the training data for the CGCF itself did not have a substantial influence.

Table 4.4 Comparative Results of CGCF Trained on Different Datasets and Systematic Search Algorithm *Diamond Energy* Using MM3 Calculations (kcal/mol).

Molecule	CGCF_Diamond_MM3	CGCF_ISO17_MM3	CGCF_qm9_MM3	CGCF_drugs_MM3	Diamond_RDKit_MM3
c4	3.90	3.09	3.09	3.09	3.086
c7n	7.91	7.86	7.86	7.09	6.251
c8a	13.82	13.06	13.06	13.02	13.016
c12p	29.50	29.50	29.50	29.50	32.78
c15helix	29.93	26.17	26.17	25.14	23.543
c17l	19.68	22.46	22.46	20.22	16.79
e12	42.81	43.37	43.37	40.35	42.805
e27	3.51	3.51	3.51	2.22	2.51
c20n	24.31	27.87	27.87	24.34	–
c39n	52.36	49.68	49.68	47.15	–
e38	25.59	25.27	25.27	26.97	–
e39	46.83	44.01	44.01	43.30	–
e40	145.81	147.28	147.28	145.92	–

The table, Table 4.5, displays the results of CGCF trained on different datasets (represented in the first four columns from the left) and the systematic search algorithm *Diamond Energy*

(depicted in the final column) using OPLS4 calculations. Molecules **c20n**, **c39n**, **e38**, **e39**, and **e40**, which were previously beyond the scope of the *Diamond Energy* algorithm, are not included. Generally, the *Diamond Energy* algorithm identified the lowest energy values more frequently compared to the CGCF model's results. However, the ANOVA test indicates that the differences between these datasets are within acceptable limits, with an F-value of 0.095 and a P-value of 0.983. This suggests that the results from the CGCF machine learning models are within a reasonable range. Additionally, the training data appears to have a minimal impact on the results.

Table 4.5 Comparative Results of CGCF Trained on Different Datasets and Systematic Search Algorithm Diamond Energy Using OPLS4 Calculations (kcal/mol).

Molecule	CGCF_Diamond_OPLS4	CGCF_ISO17_OPLS4	CGCF_qm9_OPLS4	CGCF_drugs_OPLS4	Diamond_RDKit_OPLS4
c4	2.79	2.79	2.79	3.27	3.272
c7n	5.17	5.17	5.19	5.65	4.672
c8a	3.86	3.80	3.86	4.48	3.968
c12p	2.56	7.84	7.84	7.84	2.564
c15helix	3.24	2.87	2.31	7.38	2.017
c17l	12.89	12.86	14.03	12.70	10.464
e12	31.63	31.63	33.23	31.63	33.938
e27	0.13	0.06	1.92	0.75	-10.919
c20n	15.09	14.92	15.44	16.00	–
c39n	22.70	25.22	23.84	24.51	–
e38	3.28	0.94	0.94	1.81	–
e39	3.95	5.05	5.60	13.49	–
e40	81.53	84.81	85.10	86.87	–

In addition to the lowest energy value identified, other criteria like the number of conformers found within the energy ranges of 3 kcal/mol and 5 kcal/mol are also presented below. It should be noted that these numerical terms might not be very reliable or accurate at this stage. For the machine learning models used in this study, the user must choose in advance how many conformers to generate. This uncertainty is actually not uncommon in real-world conformer search tasks, as users do not know beforehand how many conformers will be generated. In this study, the default setting is to generate a total of 25 conformers.

Table 4.6 shows the number of conformers found within 3 kcal/mol of the lowest energy value range for all CGCF pipelines using different training datasets, and the results of the *Diamond Energy* using the RDKit package to add hydrogens before the minimization method. All processes were conducted using the MM3 force field. Some molecules, such as **c20**, **c39n**, **e38**, **e39**, and **e40**, were previously outside the range of the *Diamond Energy* algorithm. Consequently, data are missing in the final column, which represents the results of *Diamond Energy*.

Table 4.6 The count of conformers within 3 kcal/mol of the lowest energy using various methods with MM3 calculations.

Molecule	Diamond_CGCF_MM3	Drugs_CGCF_MM3	Iso17_CGCF_MM3	Qm9_CGCF_MM3	Diamond Energy_MM3
c4	25	25	25	25	3
c7n	16	21	18	18	39
c8a	13	10	14	10	17
c12p	3	13	21	7	9
c15helix	5	3	8	1	90
c17l	1	8	9	5	881
e12	6	7	15	9	9
e27	3	5	11	4	20
c20n	3	9	9	10	–
c39n	1	2	4	3	–
e38	4	17	10	8	–
e39	2	1	1	2	–
e40	7	3	5	5	–

To visualize the results of Table 4.6, a dotted line chart is drawn in Figure 4.7. It appears that for flexible molecules like **c15helix** and **c17l**, the CGCF model is not as robust as systematic search methods for finding low-energy conformers within 3 kcal/mol of the lowest energy value. While in some cases there are fluctuations between different groups, it seems that the performance of the CGCF model varies with different training sets, yet there is no obvious trend indicating which dataset could provide the best guidance. To further understand why it is this result, more work is needed to delve into the details of how this machine learning model functions and its interpretability in the future.

The following Table 4.7 lists the conformers found within 5 kcal/mol of the lowest energy value using different methods with the MM3 force field. Similarly, to better analyze and compare these results, the data are visualized in Figure 4.8. The same trend observed in the 3 kcal/mol conformer count is evident here. For more complex molecules like **c7n**, **c15helix** and **c17l**, the CGCF method may not be robust enough to explore the low-energy space thoroughly. However, it can still be a good choice if users need low-energy conformers for further research, provided that a complete conformer search result is not required.

The same comparison process was conducted using OPLS4 force field calculations as well. Table 4.8 shows the results of the count of conformers within 3 kcal/mol of the lowest energy obtained using various methods with OPLS4 calculations. It appears that for flexible molecules such as **c7n**, **c15helix** and **c17l**, the CGCF model is not as robust as systematic search methods in finding low-energy conformers within 3 kcal/mol of the lowest energy value. This result is visualized in Figure 4.9.

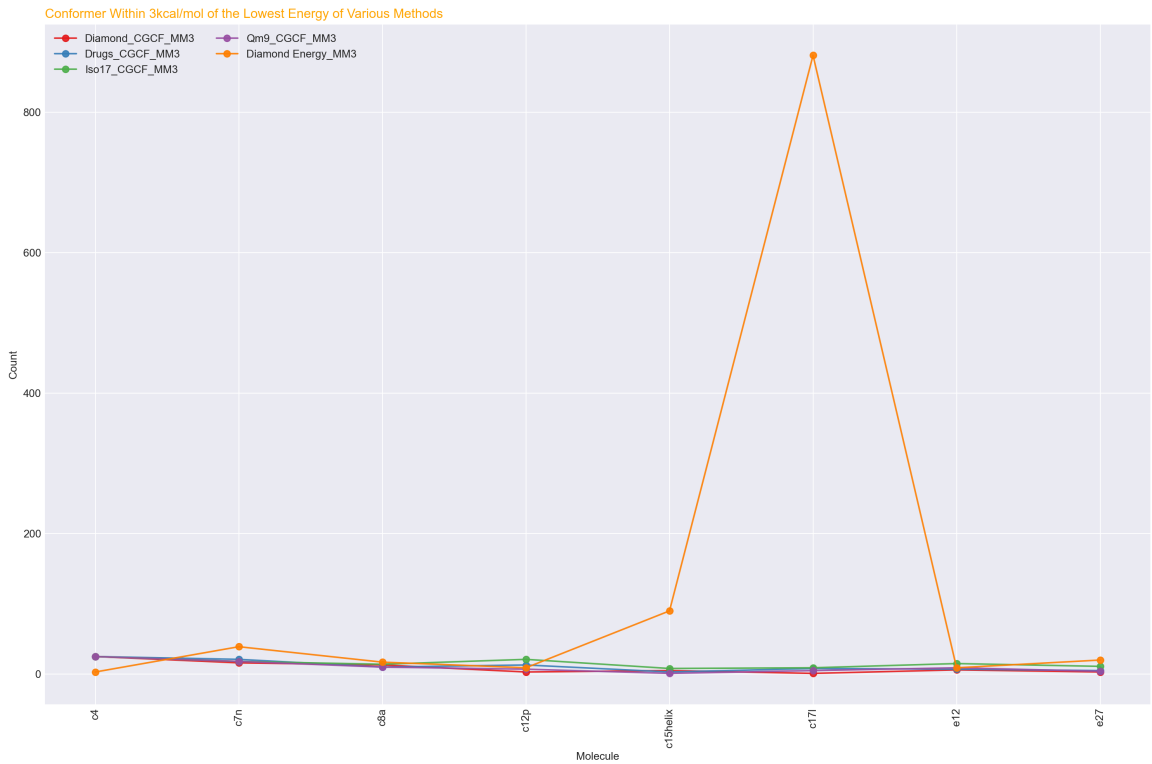


Figure 4.7 The results from Table 4.6, showing the count of conformers within 3 kcal/mol of the lowest energy obtained using various methods with MM3 calculations.

Table 4.7 The count of conformers within 5 kcal/mol of the lowest energy using various methods with MM3 calculations.

Molecule	Diamond_CGCF_MM3	Drugs_CGCF_MM3	Iso17_CGCF_MM3	Qm9_CGCF_MM3	Diamond Energy_MM3
c4	25	25	25	25	3
c7n	22	24	25	24	67
c8a	21	20	19	14	23
c12p	3	16	21	7	9
c15helix	12	5	17	7	395
c17l	6	18	15	11	2873
e12	6	11	17	13	9
e27	5	9	17	8	41
c20n	8	14	17	15	–
c39n	4	7	8	5	–
e38	12	23	21	20	–
e39	2	1	1	2	–
e40	10	5	8	11	–

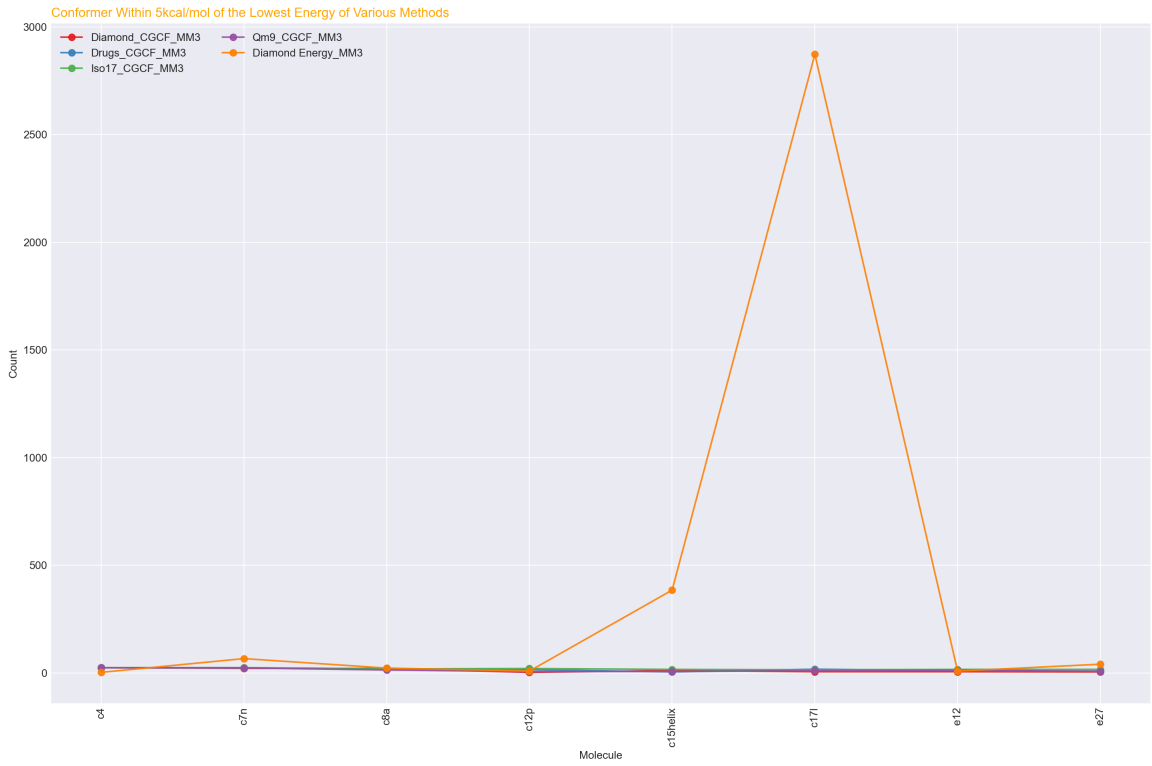


Figure 4.8 Visualization of the results from Table 4.7, showing the count of conformers within 5 kcal/mol of the lowest energy obtained using various methods with MM3 calculations.

Table 4.8 The count of conformers within 3 kcal/mol of the lowest energy using various methods with OPLS4 calculations.

Molecule	qm9_CGCF_OPLS4	iso17_CGCF_OPLS4	drugs_CGCF_OPLS4	diamond_CGCF_OPLS4	Diamond Energy_OPLS4
c4	25	25	25	25	3
c7n	19	25	21	24	55
c8a	12	19	13	14	22
c12p	14	2	20	3	9
c15helix	2	7	7	3	195
c17l	14	10	9	11	3158
e12	5	18	5	8	5
e27	4	5	4	4	18
c20n	7	13	9	6	–
c39n	5	5	4	5	–
e38	13	11	9	22	–
e39	2	3	3	1	–
e40	3	4	5	1	–

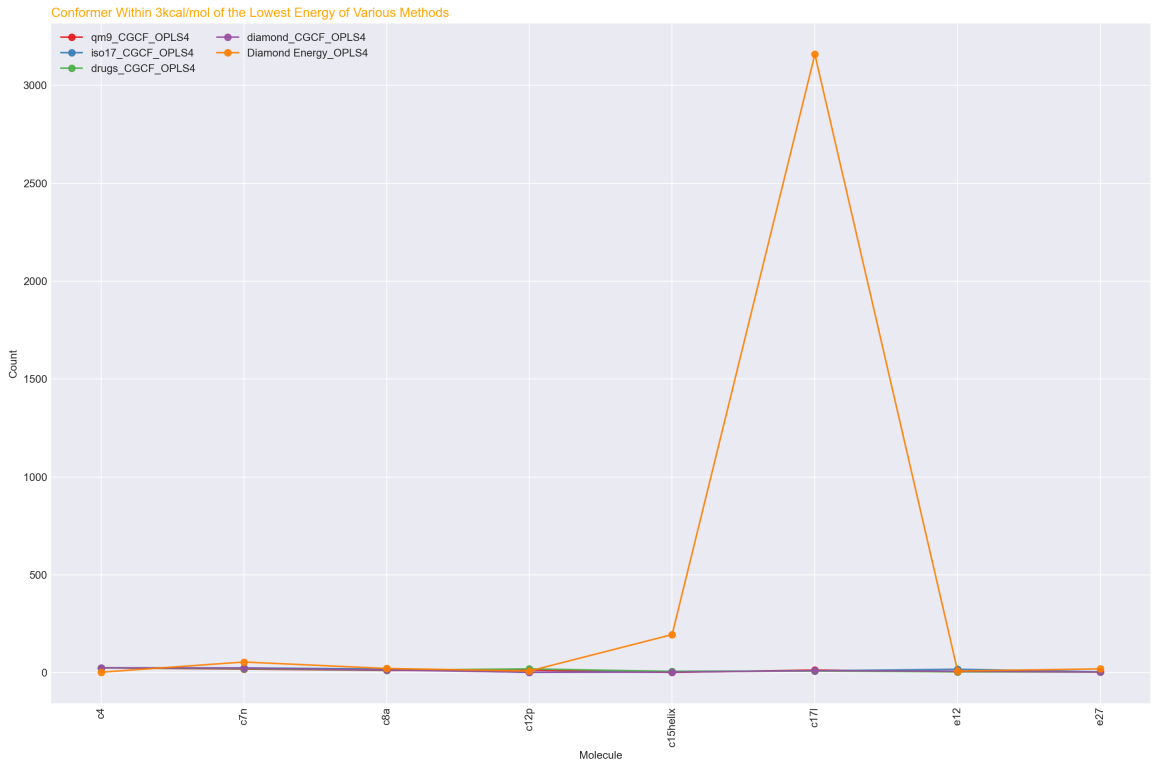


Figure 4.9 The results from Table 4.8, showing the count of conformers within 3 kcal/mol of the lowest energy obtained using various methods with OPLS4 calculations.

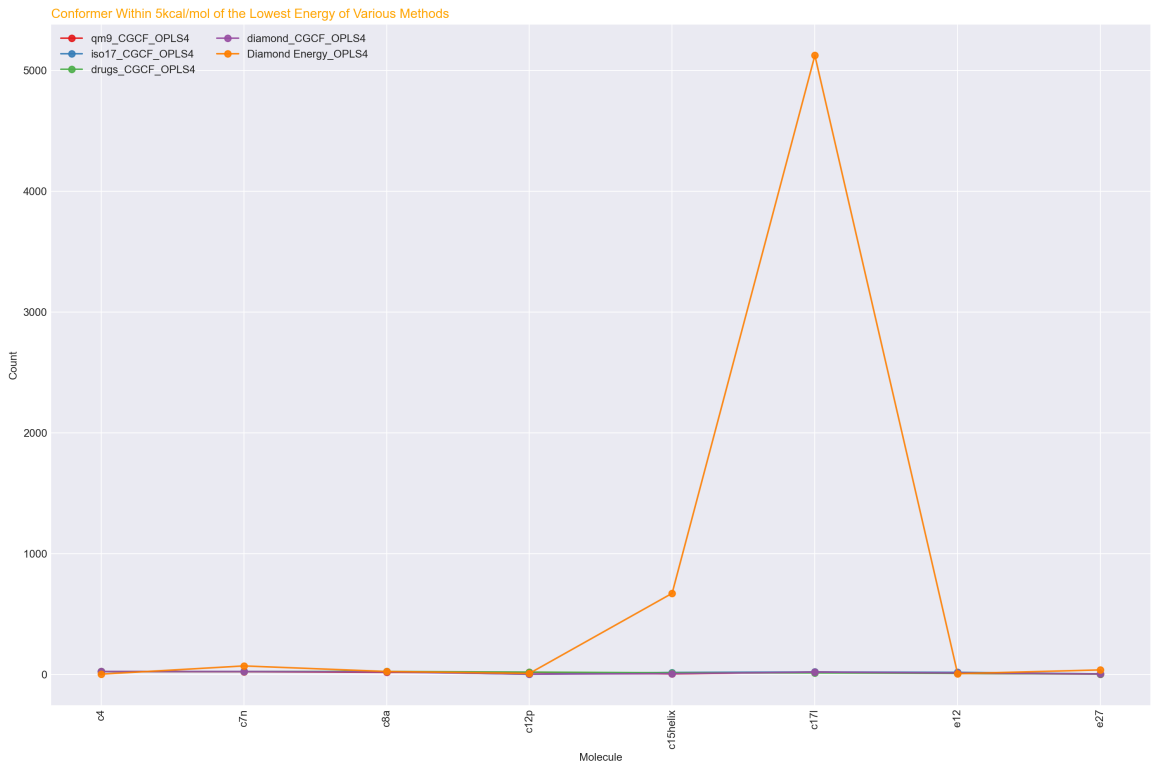


Figure 4.10 Visualization of the results from Table 4.9, showing the count of conformers within 5 kcal/mol of the lowest energy obtained using various methods with OPLS4 calculations.

The results of counting conformers within 5 kcal/mol of the lowest energy, obtained using various methods alongside OPLS4 calculations, are shown in Table 4.9 below. Figure 4.10 displays a dotted line chart of these results from Table 4.9. It is evident that for flexible and complex molecules like **c7n**, **c15helix**, **c17l** and **e27**, the CGCF method may not be able to thoroughly explore the low-energy space within 5 kcal/mol of the lowest energy range.

Table 4.9 The count of conformers within 5 kcal/mol of the lowest energy obtained using various methods with OPLS4 calculations.

Molecule	qm9_CGCF_OPLS4	iso17_CGCF_OPLS4	drugs_CGCF_OPLS4	diamond_CGCF_OPLS4	Diamond Energy_OPLS4
c4	25	25	25	25	3
c7n	23	25	25	25	71
c8a	18	24	25	21	25
c12p	18	2	20	3	9
c15helix	5	17	14	7	672
c17l	21	21	13	20	5125
e12	9	18	9	13	7
e27	4	5	4	5	38
c20n	12	17	17	14	–
c39n	5	8	8	11	–
e38	22	16	21	25	–
e39	2	4	4	1	–
e40	6	5	7	1	–

In conclusion, data obtained using different force fields, including MMFF94, MM3, and OPLS4, have been analyzed for the lowest energy values identified, the conformers count within 3 kcal/mol and 5 kcal/mol spaces of the lowest energy. The findings suggest that varying training data from sources such as *Diamond Energy* output, ISO17, QM9, DRUGs, do not significantly influence the performance of the CGCF machine learning model.

The *Diamond Energy* data were derived from a previous study, with output structures taken from .sdf files discussed in earlier chapters. The remaining test data were generated by using the well-trained CGCF model of these datasets (ISO17, QM9, DRUGs) by the CGCF author, to conduct geometry generation for the test molecules. When compared with the *Diamond Energy* algorithm, the results indicate that although the CGCF can identify some low-energy conformers, its performance is not yet sufficient to compete with traditional algorithms in terms of both the lowest energy values identified and the number of conformers found near the lowest energy space.

However, the CGCF does offer an advantage for certain molecules that are beyond the scope of traditional algorithms, such as those with an excessive number of rotatable bonds or highly complex structures. It can serve as an alternative method for conformer identification without bearing the time costs associated with traditional conformer search processes.

Variations in Training Data (Diamond, QM9, Drugs) and Their Impact on the Output of GeoMol Model Table 4.10 presents the results of conformers generated by GeoMol when trained on various datasets and subsequently energy-minimized using the RDKit MMFF94 force field (Table 4.10). The first column lists the molecular names, while the subsequent columns display results corresponding to different training datasets: *Diamond Energy* data, QM9, and drugs.

It can be seen from Table 4.10 that when using Diamond data to train GeoMol, the results do not perform well on molecules **c20n** and **c39n**. This may be due to the absence of these molecules in the training data. Apart from this, for the other test cases, the three pipelines do not exhibit a large gap. An ANOVA test was conducted to determine if there is a statistically significant gap between the groups (GeoMol_Diamond_MMFF94, GeoMol_qm9_MMFF94, GeoMol_drugs_MMFF94). The results are as follows: F-value: 0.2788, p-value: 0.7594. The ANOVA test results suggest that the differences between these groups are acceptable, and there is no significant gap in the results from different training sets for the GeoMol model.

Table 4.10 The Lowest Energy Identified for Test Molecules of GeoMol Trained on Different Datasets Using MMFF94 Calculations.

Molecule	GeoMol_Diamond_MMFF94	GeoMol_qm9_MMFF94	GeoMol_drugs_MMFF94
c12p	47.28	47.28	47.28
c15helix	23.01	22.74	22.39
c17l	1.68	−3.35	−4.32
c20n	4.44	−2.74	−3.12
c39n	40.41	8.78	2.82
c4	−5.08	−5.08	−5.08
c7n	−4.05	−4.85	−4.85
c8a	8.29	8.29	8.39

The Table 4.11 presents the results of conformers generated by GeoMol when trained on various datasets and subsequently energy-minimized using the MacroModel MM3 force field. The first column lists the molecular names, while the subsequent columns display results corresponding to different training datasets: *Diamond Energy* data, QM9, and drugs. The final column shows the lowest energy identified by *Diamond Energy* for comparison with the traditional conformer search process. The molecules **c20n** and **c39n** exceed the limits of *Diamond Energy*, resulting in missing data. It can be observed that the lowest energy values found are actually very close among these groups. An ANOVA test was conducted with an F-value of 0.017 and a p-value of 0.997, supporting that these groups' difference is fall into normal fluctuation range.

Table 4.11 The Lowest Energy Identified for Test Molecules of GeoMol Trained on Different Datasets Using MM3 Calculations.

Molecule	GeoMol_Diamond_MM3	GeoMol_qm9_MM3	GeoMol_drugs_MM3	Diamond Energy_MM3
c4	3.09	3.09	3.09	3.09
c7n	7.14	7.09	7.09	6.25
c8a	13.06	13.02	13.02	13.02
c12p	29.50	29.50	29.50	32.78
c15helix	28.96	26.65	26.43	23.54
c17l	22.23	21.52	20.24	16.79
c20n	28.94	24.57	25.17	–
c39n	59.02	56.16	53.96	–

Table 4.12 below presents the results of conformers generated by GeoMol when trained on various datasets and subsequently energy-minimized using the MacroModel OPLS4 force field. The first column lists the molecular names, while the subsequent columns display results corresponding to different training datasets: *Diamond Energy* data, QM9, and drugs. Similarly, the *Diamond Energy* results are generally better than those of the GeoMol pipelines, but there is no significant difference between these results, as indicated by the ANOVA test with an F-value of 0.037 and a p-value of 0.963, showing these groups' results are close.

Table 4.12 The Lowest Energy Identified for Test Molecules of GeoMol Trained on Different Datasets Using OPLS4 Calculations.

Molecule	GeoMol_Diamond_OPLS4	GeoMol_qm9_OPLS4	GeoMol_drugs_OPLS4	Diamond Energy_OPLS4
c4	2.79	2.79	2.79	3.27
c7n	5.19	5.17	5.17	4.67
c8a	4.03	3.97	3.83	3.97
c12p	2.56	7.84	8.29	2.56
c15helix	5.43	4.29	5.82	2.02
c17l	15.08	14.06	12.91	10.46
c20n	16.46	15.68	15.84	–
c39n	29.95	31.63	33.42	–

The number of conformers within 3 kcal/mol and 5 kcal/mol of the lowest energy value is also compared for both MM3 and OPLS4 force fields for the GeoMol models with different training datasets and the traditional search method, *Diamond Energy*.

Table 4.13 and Figure 4.11 displays the count for different GeoMol pipelines and *Diamond Energy* within the 3 kcal/mol range of the lowest energy value using the MM3 force field. The results indicate that *Diamond Energy* still has a greater advantage for complex molecules such as **c7n**, **c15helix**, and **c17l** within the low energy space. However, this might not be accurate, as the number of conformers for the machine learning model must be predefined at the outset, and it is not known in advance before the conformer search begins. A similar

trend can be observed in the results for the 5 kcal/mol range within the lowest energy space (referenced in Table 4.14 and Figure 4.12). While there is still a gap between GeoMol and the traditional method, the data itself does not significantly affect GeoMol's performance.

Table 4.13 Showing the count of conformers within 3 kcal/mol of the lowest energy obtained using various GeoMol methods and *Diamond Energy* method with MM3 calculations.

Molecule	qm9_GeoMol_MM3	diamond_GeoMol_MM3	drugs_GeoMol_MM3	Diamond Energy_MM3
c4	19	19	20	3
c7n	16	10	19	39
c8a	15	12	13	17
c12p	5	6	4	9
c15helix	7	4	8	90
c17l	7	2	4	881
c20n	3	4	5	–
c39n	3	1	7	–

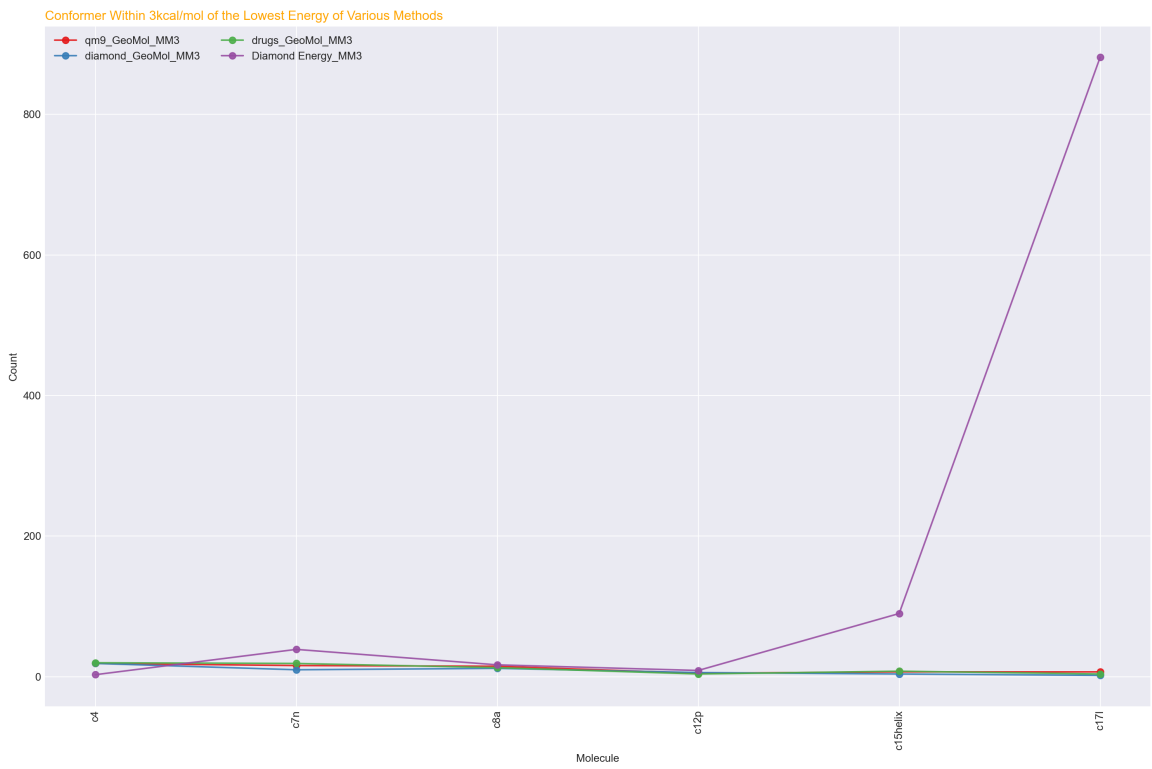


Figure 4.11 The results from Table 4.13, showing the count of conformers within 3 kcal/mol of the lowest energy obtained using various methods with MM3 calculations.

The same comparison process has also been checked using the OPLS4 force field. The subsequent tables (Table 4.15 and Table 4.16) and figures (Figure 4.13 and Figure 4.14) present the count of conformers within 3 kcal/mol and 5 kcal/mol of the lowest energy range

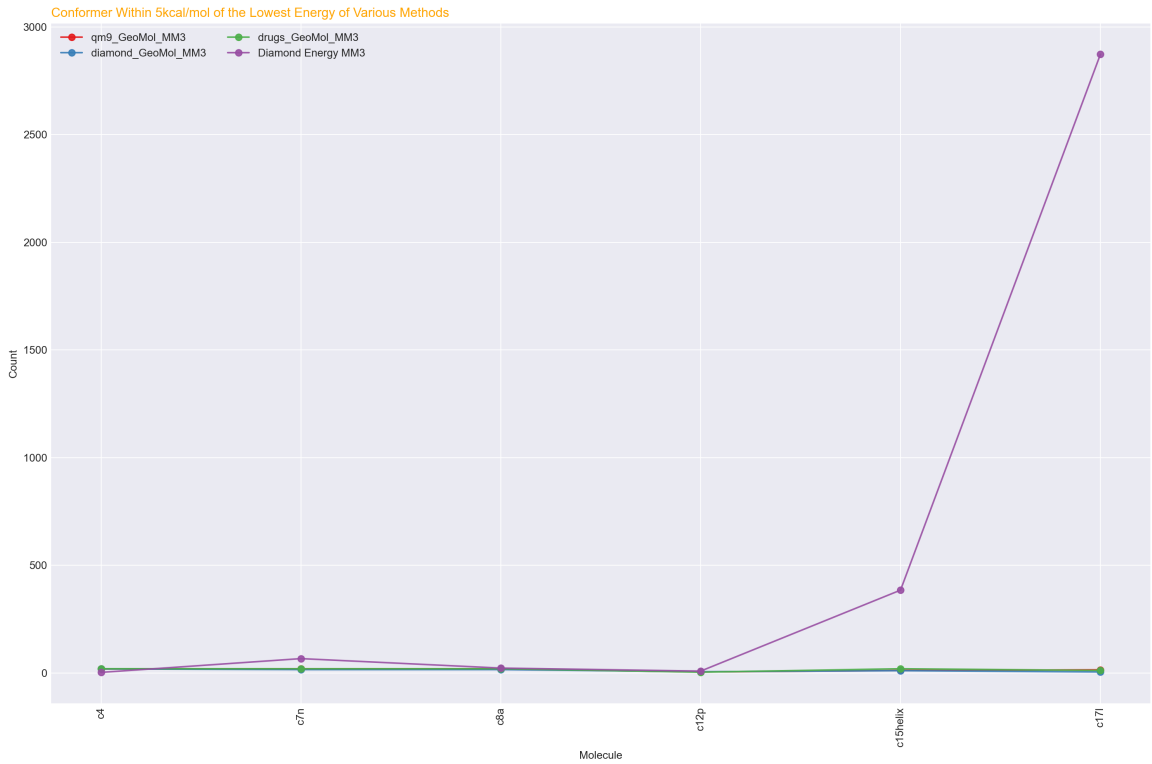


Figure 4.12 The results from Table 4.14, showing the count of conformers within 5 kcal/mol of the lowest energy obtained using various methods with MM3 calculations.

Table 4.14 The count of conformers within 5 kcal/mol of the lowest energy obtained using various GeoMol methods and *Diamond Energy* method with MM3 calculations.

Molecule	qm9_GeoMol_MM3	diamond_GeoMol_MM3	drugs_GeoMol_MM3	Diamond Energy_MM3
c4	19	19	20	3
c7n	19	16	20	67
c8a	19	16	20	23
c12p	5	6	4	9
c15helix	12	11	20	385
c17l	15	5	12	2873
c20n	4	5	10	–
c39n	5	1	9	–

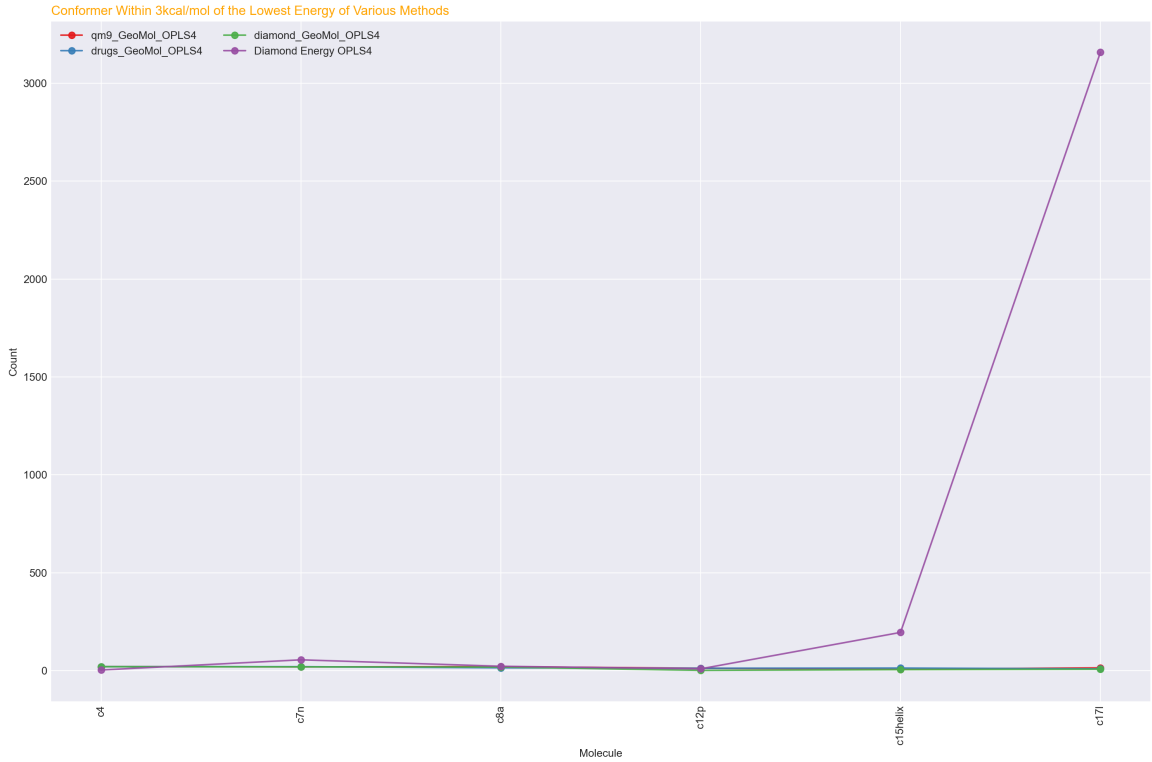


Figure 4.13 The results of the count of conformers within 3 kcal/mol of the lowest energy obtained using various methods with OPLS4 calculations.

Table 4.15 The count of conformers within 3 kcal/mol of the lowest energy obtained using various GeoMol methods and *Diamond Energy* method with OPLS4 calculations.

Molecule	qm9_GeoMol_OPLS4	drugs_GeoMol_OPLS4	Diamond_GeoMol_OPLS4	Diamond Energy_OPLS4
c4	20	20	20	3
c7n	19	20	19	55
c8a	20	14	18	22
c12p	12	12	1	9
c15helix	7	13	5	195
c17l	14	9	7	3158
c20n	3	6	7	–
c39n	2	8	7	–

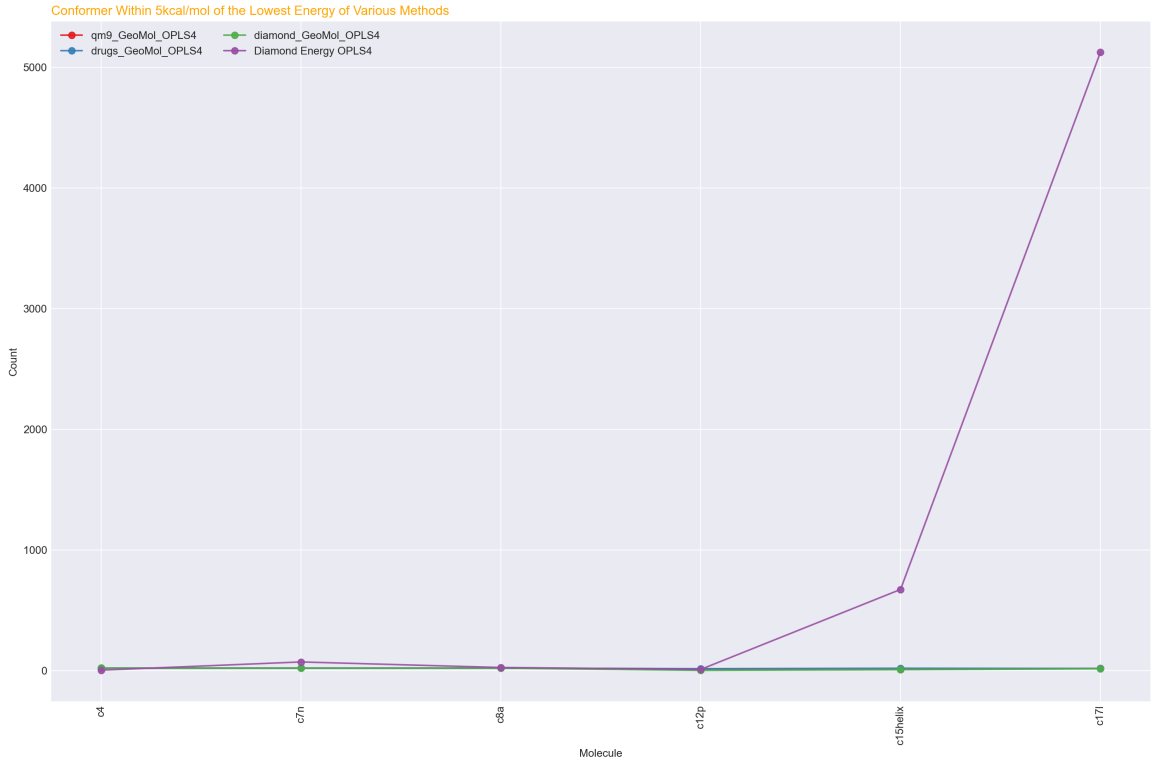


Figure 4.14 The results of the count of conformers within 5 kcal/mol of the lowest energy obtained using various methods with OPLS4 calculations.

Table 4.16 The count of conformers within 5 kcal/mol of the lowest energy obtained using various GeoMol methods and *Diamond Energy* method with OPLS4 calculations.

Molecule	qm9_GeoMol_OPLS4	drugs_GeoMol_OPLS4	Diamond_GeoMol_OPLS4	Diamond Energy_OPLS4
c4	20	20	20	3
c7n	20	20	20	71
c8a	20	20	20	25
c12p	13	15	1	9
c15helix	9	19	7	672
c17l	17	17	15	5125
c20n	11	13	10	–
c39n	17	10	7	–

for GeoMol models and *Diamond Energy*. It shows that the results of GeoMol models are still not competitive with *Diamond Energy* and the training data itself does not affect the GeoMol model as expected.

As with the results of CGCF, the training data do not significantly influence the performance of the GeoMol model and there is still some way to go for these models to replace traditional algorithms like *Diamond Energy*. However, for some molecules that exceed the limits of *Diamond Energy*, these models are good alternatives for finding low-energy structures.

Diamond Training Data on ConfVAE Model VS *Diamond Energy* Algorithm The ConfVAE model has been trained exclusively with the *Diamond Energy* dataset, as a version trained with a different dataset has not been provided by the ConfVAE project. The lowest energies identified by ConfVAE and *Diamond Energy* using MM3 and OPLS4 force fields are listed in the following tables (Table 4.17 and Table 4.18). Similar to the results from CGCF and GeoMol, it can be seen that these lowest energies identified by ConfVAE are actually very close to those identified by *Diamond Energy*. Although traditional algorithms still hold a slight edge in most cases—with the *Diamond Energy* values being lower—for some molecules that exceed the limits of *Diamond Energy*, ConfVAE can be used to find low-energy structures.

Table 4.17 Comparative Lowest Energy Identified Results of ConfVAE and *Diamond Energy* Using MM3 Calculations.

Molecule	ConfVAE_MM3	*Diamond Energy_MM3*
c4	3.09	3.09
c7n	7.09	6.25
c8a	13.00	13.02
c12p	29.50	32.78
c15helix	23.61	23.54
c17l	19.33	16.79
e12	48.29	42.81
e27	2.36	2.51
c20n	28.23	–
c39n	61.07	–
e38	29.65	–
e39	9674.85	–
e40	144.56	–

Table 4.18 Comparative Lowest Energy Identified Results of ConfVAE and *Diamond Energy* Using OPLS4 Calculations.

Molecule	ConfVAE_OPLS4	*Diamond Energy_OPLS4*
c4	2.79	3.27
c7n	5.17	4.67
c8a	3.80	3.97
c12p	7.84	2.56
c15helix	2.11	2.02
c17l	10.95	10.46
e12	33.23	33.94
e27	0.11	−10.92
c20n	16.53	–
c39n	37.59	–
e38	5.70	–
e39	1139.21	–
e40	103.52	–

The results of using ConfVAE to identify low-energy conformers within 5 kcal/mol of the lowest energy value using both MM3 and OPLS4 force fields are also listed in the following tables (Table 4.19 and Table 4.20). The ConfVAE model does not require a predefined number of outputs, but the conformers produced by this model are not as comprehensive as those from *Diamond Energy* as expected. Again, this model shows promising potential as a conformer search process and can be used as an alternative for molecules beyond the limits of traditional methods.

Table 4.19 The count of conformers within 5 kcal/mol of the lowest energy obtained using various ConfVAE method and *Diamond Energy* method with MM3 calculations.

Molecule	ConfVAE_MM3	*Diamond Energy_MM3*
c4	3	3
c7n	63	67
c8a	13	23
c12p	1	9
c15helix	113	385
c17l	1681	2873
e12	7	9
e27	6	41

Table 4.20 The count of conformers within 5 kcal/mol of the lowest energy obtained using various *ConfVAE* method and *Diamond Energy* method with OPLS4 calculations.

Molecule	ConfVAE_OPLS4	*Diamond Energy_OPLS4*
c4	3	3
c7n	70	71
c8a	19	25
c12p	7	9
c15helix	256	672
c17l	2155	5125
e12	5	7
e27	23	38

GraphDG Results GraphDG has not been successfully run and trained so far. The model was developed earlier, and many of the package's dependencies it utilizes are from earlier versions, which has resulted in significant time spent resolving environment conflicts. Some issues remain unresolved. Becoming familiar with Linux system operations and gaining more experience in building machine learning models might address these challenges. As such, a deeper exploration of this model may be reserved for the future when the necessary expertise is acquired.

Conclusions Since the *Diamond Energy* algorithm is theoretically level-independent, while other machine learning (ML) models rely on their quantum mechanics (QM) level data. Also, extending the scope of *Diamond Energy* might be time-consuming due to the need to modify and test the code. It is then suggest to use the *Diamond Energy* easy to get low energy data as training set to teach the ML models to do conformer search task.

This chapter attempts to use ML models to learn the pattern of low energy geometries similar to diamond lattice, with the hope that it could be easily adapted for fast conformer search ML models.

The results so far have not shown that the *Diamond Energy* data can significantly improve the performance of these ML-based models. This might due to the current study process have not been able to fully utilize the *Diamond Energy*'s information from these training data as expected. Nevertheless, *Diamond Energy* data are easy to acquire, and the diamond data's results so far do not show a large gap with that of the QM level data. It still demonstrates the potential of utilizing this data to quickly identify patterns in low-energy conformer geometries and then use this information for a fast and general conformer search.

More work could be done, but there is not enough time currently, as it requires a deeper understanding of the mechanisms of different machine learning models. Only after this understanding can we hope to improve or redesign models to learn the information in *Diamond Energy* data to achieve the initial goal. Other models, such as those based on reinforcement learning, might be more adept at guiding the generation of low-energy conformer distributions following diamond lattice pattern. This area remains open for further exploration.

Additionally, results to date also suggest that ML-based models are not robust enough to compete with traditional conformer search methods. However, they can still be a good choice for molecules that are beyond the scope of traditional methods.

Exploring the Potential for Accelerated Conformer Minimization: Machine Learning Models vs. PRDG Evaluate the capability of identifying stable molecules using two minimization models, ConfGF and Geodiff. The comparison process is conducted as follows: *Diamond Energy* test molecules are processed through three minimization procedures—ConfGF minimization, Geodiff minimization, and PRCG minimization in MacroModel. Since the two models only provide output of minimized structures, the accurate energy values for them will be determined using MacroModel's single point energy calculation with both MM3 and OPLS4 force fields. The final minimization values will then be compared.

Below are the results derived from the GeoDiff model, which results come from a test set processed using a well-trained GeoDiff model provided by the author. The procedure involves inputting a test molecule's random 3D geometry, after which GeoDiff outputs a stable conformer. ConfGF was also used directly from a well-trained model provided by the author. For ConfGF, there is a considerable amount of final data, approximately 113 GB. This represents a disadvantage of ConfGF, as the process requires an excessive amount of memory.

The tables (Table 4.21 and Table 4.22) show the lowest energy identified by GeoDiff, ConfGF, and Diamond Energy-PRCG minimization using both MM3 and OPLS4 force fields. Some molecules posed problems, resulting in missing data in GeoDiff and ConfGF. Furthermore, due to ConfGF's substantial demand on storage space and GPU memory, the program often terminates itself when processing most molecules, as it runs out of the required GPU memory.

Table 4.21 Comparative Results of ConfVAE and Diamond-PRCG Using MM3 Calculations.

Molecule	**ConfGF_MM3**	**GeoDiff_MM3**	***Diamond-PRCG_MM3***
e1	9.10	9.10	9.10
e10	17.79	22.81	14.41
e11	13.65	15.60	13.65
e12	45.95	56.90	42.81
e13	16.23	16.23	13.22
e14	17.80	31.60	14.54
e15	19.85	34.83	18.02
e16	34.81	108.41	22.15
e17	18.41	57.99	16.64
e18	16.23	18.50	13.22
e19	12.94	90.80	12.94
e2	12.29	–	12.29
e20	13.65	–	13.65
e21	13.65	15.60	13.65
e22	12.48	12.48	11.96
e23	15.76	21.04	14.98
e24	15.65	53.43	14.92
e25	15.23	22.53	12.08
e26	17.86	185.99	17.18
e27	3.21	13.60	2.51
e28	19.78	68.60	18.26
e29	37.72	41.08	16.95
e3	9.59	24.89	9.59
e30	17.92	19.90	14.84
e31	17.92	18.19	14.84
e32	16.62	17.52	13.71
e33	21.94	–	20.81
e34	31.28	173.51	29.61
e35	62.17	–	43.13
e36	212.58	110.72	73.59
e37	150.03	164.52	121.24
e4	9.59	9.59	9.59
e5	9.59	24.89	9.59

Table 4.21 continued from previous page

Molecule	**ConfGF_MM3**	**GeoDiff_MM3**	***Diamond-PRCG_MM3***
e6	21.98	48.80	20.04
e7	12.04	12.61	12.04
e8	15.75	34.97	13.01
e9	32.37	–	32.37
c10n	–	15.64	6.55
c12p	–	53.57	32.78
c12p0	–	53.56	32.78
c12p1	–	82.53	38.75
c15	–	291.11	24.71
c15a	–	111.42	24.57
c15b	–	83.94	24.60
c15c	–	56.87	24.61
c15d	–	123.71	24.69
c15e	–	53.34	24.48
c15f	–	62.31	24.66
c15g	–	37.50	24.70
c15helix	–	133.28	23.54
c16	–	114.70	20.76
c17	–	259.68	46.18
c17l	–	25.22	16.79
c18	–	61.16	28.95
c18n	–	58.75	17.84
c19	–	45.38	34.43
c21	–	112.30	34.12
c26	–	526.46	54.27
c4	3.90	–	3.09
c5	–	11.53	4.30
c5n	4.14	4.14	–
c6	–	21.71	8.18
c6n	6.83	5.20	–
c7	–	26.50	11.63
c7c	–	13.04	8.62
c7n	8.07	–	6.25

Table 4.21 continued from previous page

Molecule	**ConfGF_MM3**	**GeoDiff_MM3**	***Diamond-PRCG_MM3***
c8	–	25.94	16.27
c8a	–	27.97	13.02
c8s	–	30.41	13.00
c8si	–	75.05	13.00
c9	–	29.64	17.46
c9n	–	99.38	8.36
c9s	–	66.61	14.09

Table 4.22 The lowest energy values identified by GeoDiff, ConfGF, and Diamond-PRCG minimization using OPLS4 calculations.

Molecule	**ConfGF_OPLS4**	**GeoDiff_OPLS4**	***Diamond_OPLS4***
e1	5.66	5.66	5.66
e10	11.06	13.64	7.44
e11	24.98	–	−5.93
e12	35.91	41.34	33.94
e13	7.89	7.89	5.41
e14	10.76	32.47	8.29
e15	7.93	11.23	7.93
e16	8.25	506.45	6.07
e17	6.39	27.78	6.33
e18	7.90	15.41	5.41
e19	6.21	–	−7.07
e2	5.32	5.32	5.32
e20	24.98	24.98	−5.93
e21	24.98	–	−5.93
e22	2.10	2.74	2.10
e23	1.36	6.17	1.36
e25	62.28	–	−8.18
e26	0.15	1772.86	−11.70
e27	0.13	30.50	−1.61
e28	4.85	130.49	−10.92
e29	5.18	5.93	4.16

Table 4.22 continued from previous page

Molecule	**ConfGF_OPLS4**	**GeoDiff_OPLS4**	***Diamond-PRCG_OPLS4***
e3	–	–	−10.07
e30	7.36	7.77	−5.90
e31	7.36	14.20	2.36
e32	7.06	7.06	2.36
e33	5.39	933.42	3.59
e34	0.69	974.85	4.90
e35	19.10	192.58	0.44
e36	224.53	450.23	4.49
e37	81.21	172.96	37.81
e38	33.17	33.17	69.37
e4	–	–	−10.07
e5	–	–	−10.07
e6	1.34	–	1.34
e7	0.04	–	−1.88
e8	3.26	3.26	1.81
e9	20.40	29.04	20.40
c10n	–	15.64	6.55
c12p	–	–	2.56
c12p0	–	23.05	2.56
c12p1	–	294.87	12.50
c15	–	44.88	4.12
c15a	–	–	3.77
c15b	–	379.02	3.95
c15c	–	30.85	4.03
c15d	–	920.69	3.92
c15e	–	10.96	3.64
c15f	–	935.86	3.69
c15g	–	31.42	3.93
c15helix	–	937.42	2.02
c16	–	67.47	6.76
c17	–	2539.51	29.62
c17l	–	13.93	10.94
c18	–	23.08	2.10

Table 4.22 continued from previous page

Molecule	**ConfGF_OPLS4**	**GeoDiff_OPLS4**	***Diamond-PRCG_OPLS4***
c18n	–	16.51	11.57
c19	–	40.68	14.14
c21	–	787.49	1.78
c26	–	4874.56	25.18
c4	–	3.27	2.79
c5	–	–	−10.93
c5n	–	3.44	3.44
c6	–	11.07	−1.26
c6n	–	4.95	4.05
c7	–	5.90	−4.85
c7c	–	13.61	2.91
c7n	–	5.76	4.67
c8	–	4.33	−5.24
c8a	–	14.53	3.97
c8s	–	15.89	3.80
c8si	–	796.25	3.80
c9	–	20.44	8.92
c9n	–	897.59	5.92
c9s	–	544.78	4.48

To visualize the results, a line chart is drawn from these data as shown in Figure 4.15 and Figure 4.16. It can be observed that, even after reaching this final converged energy, the lowest energy minimized by GeoDiff and ConfGF is still far from the PRCG minimization results. These data do not support the idea that they can be used as proper minimization methods, but rather more like models to minimize some very poor structures from a starting point for further minimization.

For other criteria, such as the conformer count within the lowest energy value range of 3 kcal/mol and 5 kcal/mol, evaluations were also conducted. However, there is a very large gap between these pipelines; for ConfGF and GeoDiff, most counts are one or several, while PRCG can reach up to hundreds and thousands. The details are not shown here, but all the original data are attached in the appendix.

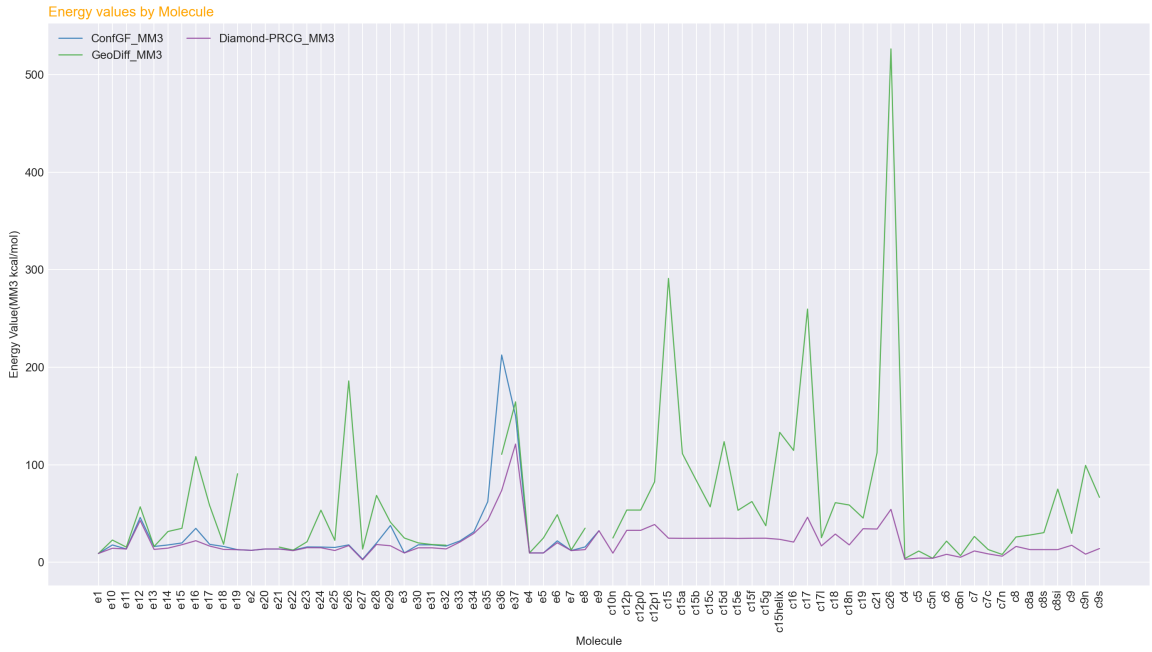


Figure 4.15 The results of the final minimization energy value of GeoDiff, ConfGF, and Diamond-PRCG using MM3.

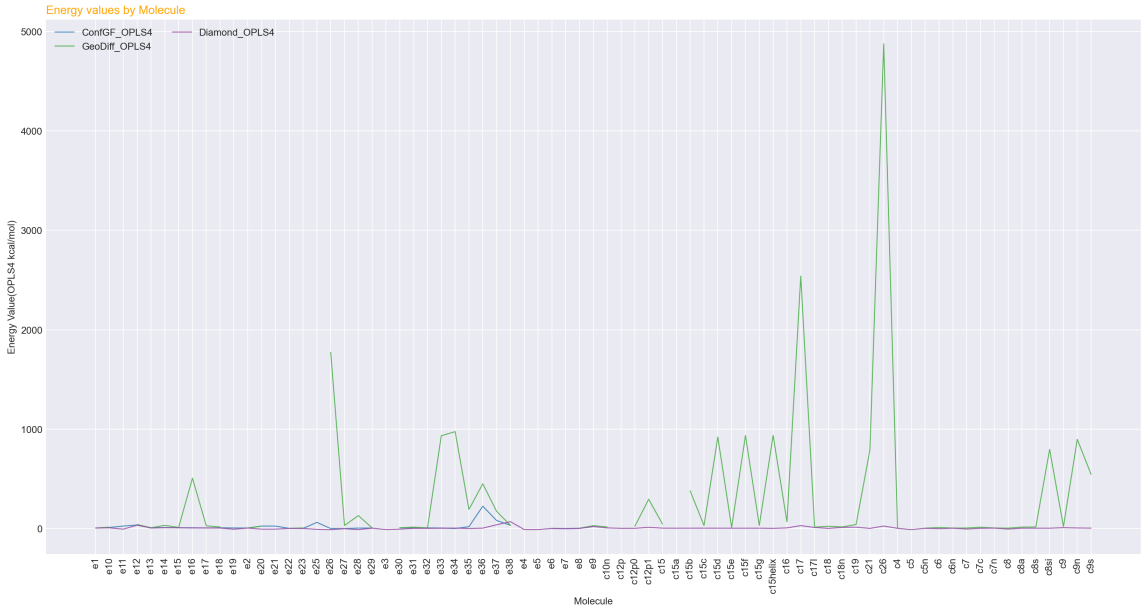


Figure 4.16 The results of the final minimization energy value of GeoDiff, ConfGF, and Diamond-PRCG using OPLS4.

The running times were also checked. ConfGF and GeoDiff took several weeks, which is significantly longer than PRCG, which only took several days. Therefore, the details are not presented here.

Conclusions Current results indicate that the objective of utilizing machine learning models to expedite the convergence of final minimized outputs in conformer search is still a distant goal. These methods do not yet match the performance of PRCG in terms of achieving the lowest energy values or computational efficiency.

Further work is necessary to refine these approaches, which presents an opportunity for continued exploration in the future.

4.2 Conclusions

The experimental sections confirmed the initial assumptions that machine learning models (for conformer search task and structure minimization task) tested in this study mostly can generate low-energy conformers quickly, which serve as starting points for further minimization or additional computational chemistry calculations. However, a challenge arises when users need to determine the number of conformers to be generated in advance. In real experiments, predicting the exact number of conformers before conducting a search is difficult. Furthermore, these models do not always guarantee comprehensive results, when comparing with traditional conformer search.

The findings in this chapter indicate that the models examined are not able to fully capture the information provided by the *Diamond Energy* data, specifically regarding the low-energy configurations that resemble a diamond lattice. It is also observed that varying the training data had little impact on the success of the conformer search. The current approach does show promise in using *Diamond Energy* data, which is easier to obtain compared with QM level low energy conformers, to quickly generate conformers. This is noteworthy because the amount of *Diamond Energy* data used for training was actually less than what is typically used in machine learning projects, and it still contributed to the results.

This segment of the work highlights the potential for more effective use of data from *Diamond Energy*, which captures the low-energy space of numerous molecules. However, achieving the primary objective of employing a model to discern low-energy configurations from *Diamond*

Energy data—and leveraging this insight for efficient conformer searches—requires further research. Moreover, a precise mathematical definition of the diamond lattice-like low-energy approximation needs to be established. This mathematical representation must encompass all features present in the *Diamond Energy* project data. Additionally, a deeper understanding of these statistical models is essential, including insights into the types of data models can learn and the extent of their learning to date. Addressing these aspects may be a task for future research endeavors.

Chapter 5

Conclusions

To fully grasp molecular properties, it is important to know all low-energy conformations. A Python program called *Diamond Energy*, which is based on the concept that the structures of local minima are similar to a diamond lattice, has been developed. This program can then use integer arithmetic for both geometry calculations and energy evaluations, aiming to speed up systematic searches. I conducted tests on a series of alkanes and found that *Diamond Energy* can perform systematic conformational searches on acyclic alkanes both quickly and accurately.

I expanded the use of *Diamond Energy* to saccharides including six-membered rings. This expansion was facilitated by an automated method that determines the appropriate energy evaluation parameters for different atom types.

The results from both the alkane system and saccharide-like systems demonstrate the algorithm's effectiveness in systematic conformer searches.

I also applied machine learning statistical models to the extensive data from *Diamond Energy*, aiming to analyze the geometric distribution of diamond lattice structures and to improve the conformational search method. The initial findings suggest that this approach may aid in conformer searches. However, to fully utilize a Diamond Machine Learning Model, more refinements are necessary to accurately represent the geometric distribution of *Diamond Energy*'s output. This requires a deeper understanding of how machine learning models work. Since this cannot be accomplished very soon, I plan to engage in further learning and research in the future.

In conclusion, this project validates the concept that structures of local minima closely mirror the diamond lattice, which aids in faster conformer searches.

References

[1] J.A. Pradeilles, S. Zhong, M. Baglyas, G. Tarczay, C.P. Butts, E.L. Myers, and V.K. Aggarwal. Odd–even alternations in helical propensity of a homologous series of hydrocarbons. *Nature Chemistry*, 12(5):475–480, 2020.

[2] A. S. Dreiding. Einfache molekularmodelle. *Helvetica Chimica Acta*, 42(4):1339–1344, 1959.

[3] W. L. Koltun. Precision space-filling atomic models. *Biopolymers: Original Research on Biomolecules*, 3(6):665–679, 1965.

[4] M. Turner. Ball and stick models for organic chemistry. *Journal of Chemical Education*, 48(6):407, 1971.

[5] T. L. Brown, H. E. LeMay, and B. E. Bursten. *Chemistry: The Central Science*. Pearson Educación, 2002.

[6] D. B. Boyd. The computational chemistry literature. *Reviews in Computational Chemistry*, 2:461–479, 1991.

[7] R. Langridge, T. E. Ferrin, I. D. Kuntz, and M. L. Connolly. Real-time color graphics in studies of molecular interactions. *Science*, 211(4483):661–666, 1981.

[8] R. J. Feldman, C. R. Bacon, and J. S. Cohen. Versatile interactive graphics display system for molecular modelling by computer. *Nature*, 244(5411):113–115, 1973.

[9] F. Mohamadi, N. G. Richards, W. C. Guida, R. Liskamp, M. Lipton, C. Caufield, G. Chang, T. Hendrickson, and W. C. Still. Macromodel—an integrated software system for modeling organic and bioorganic molecules using molecular mechanics. *Journal of Computational Chemistry*, 11(4):440–467, 1990.

[10] MacroModel. *Maestro*. Schrödinger, LLC, New York, NY, 2022.

[11] A. R. Leach. *Molecular modelling: principles and applications*. Pearson Education, 2001.

[12] S. G. Smith and J. M. Goodman. Assigning stereochemistry to single diastereoisomers by giao nmr calculation: The dp4 probability. *Journal of the American Chemical Society*, 132(37):12946–12959, 2010.

[13] A. Howarth, K. Ermanis, and J. M. Goodman. Dp4-ai automated nmr data analysis: straight from spectrometer to structure. *Chemical Science*, 11(17):4351–4359, 2020.

[14] A. Howarth and J. M. Goodman. The dp5 probability, quantification and visualisation of structural uncertainty in single molecules. *Chemical Science*, 13(12):3507–3518, 2022.

[15] P. Pracht, F. Bohle, and S. Grimme. Automated exploration of the low-energy chemical space with fast quantum chemical methods. *Physical Chemistry Chemical Physics*, 22(14):7169–7192, 2020.

[16] I. Kolossváry and W. C. Guida. Low mode search. an efficient, automated computational method for conformational analysis: Application to cyclic and acyclic alkanes and cyclic peptides. *Journal of the American Chemical Society*, 118(21):5011–5019, 1996.

[17] I. Kolossváry and G. M. Keserü. Hessian-free low-mode conformational search for large-scale protein loop optimization: application to c-jun n-terminal kinase jnk3. *Journal of Computational Chemistry*, 22(1):21–30, 2001.

[18] H. Satoh, T. Oda, K. Nakakoji, T. Uno, H. Tanaka, S. Iwata, and K. Ohno. Potential energy surface-based automatic deduction of conformational transition networks and its application on quantum mechanical landscapes of d-glucose conformers. *Journal of Chemical Theory and Computation*, 12(11):5293–5308, 2016.

[19] T. A. Frewen, G. Hummer, and I. G. Kevrekidis. Exploration of effective potential landscapes using coarse reverse integration. *The Journal of Chemical Physics*, 131(13), 2009.

[20] S. Maeda, Y. Harabuchi, Y. Ono, T. Taketsugu, and K. Morokuma. Intrinsic reaction coordinate: Calculation, bifurcation, and automated search. *International Journal of Quantum Chemistry*, 115(5):258–269, 2015.

[21] J. M. Goodman and M. A. Silva. Qrc: a rapid method for connecting transition structures to reactants in the computational analysis of organic reactivity. *Tetrahedron Letters*, 44(45):8233–8236, 2003.

[22] S. A. Trygubenko and D. J. Wales. A doubly nudged elastic band method for finding transition states. *The Journal of Chemical Physics*, 120(5):2082–2094, 2004.

[23] J. M. Carr, S. A. Trygubenko, and D. J. Wales. Finding pathways between distant local minima. *The Journal of Chemical Physics*, 122(23), 2005.

[24] D. J. Wales. Energy landscapes: some new horizons. *Current Opinion in Structural Biology*, 20(1):3–10, 2010.

[25] S. Bell and J. S. Crighton. Locating transition states. *The Journal of Chemical Physics*, 80(6):2464–2475, 1984.

[26] M. De Vivo, M. Masetti, G. Bottegoni, and A. Cavalli. Role of molecular dynamics and related methods in drug discovery. *Journal of Medicinal Chemistry*, 59(9):4035–4061, 2016.

[27] A. E. Howard and P. A. Kollman. An analysis of current methodologies for conformational searching of complex molecules. *Journal of Medicinal Chemistry*, 31(9):1669–1675, 1988.

[28] M. Vasquez, G. Nemethy, and H. A. Scheraga. Computed conformational states of the 20 naturally occurring amino acid residues and of the prototype residue -aminobutyric acid. *Macromolecules*, 16(7):1043–1049, 1983.

[29] M. Vásquez and H. A. Scheraga. Use of buildup and energy-minimization procedures to compute low-energy structures of the backbone of enkephalin. *Biopolymers: Original Research on Biomolecules*, 24(8):1437–1447, 1985.

[30] K. D. Gibson and H. A. Scheraga. Revised algorithms for the build-up procedure for predicting protein conformations by energy minimization. *Journal of Computational Chemistry*, 8(6):826–834, 1987.

[31] R. L. Ornstein and R. Rein. Energetics of intercalation specificity. i. backbone unwinding. *Biopolymers: Original Research on Biomolecules*, 18(5):1277–1291, 1979.

[32] B. E. Hingerty, S. Figueroa, T. L. Hayden, and S. Broyde. Prediction of dna structure from sequence: A build-up technique. *Biopolymers: Original Research on Biomolecules*, 28(7):1195–1222, 1989.

[33] W. C. Still and I. Galynker. Chemical consequences of conformation in macrocyclic compounds: an effective approach to remote asymmetric induction. *Tetrahedron*, 37(23):3981–3996, 1981.

[34] Y. Fukazawa, S. Usui, Y. Uchio, Y. Shiobara, and M. Kodama. Conformational study of the cembranolide diterpene denticulatolide by molecular mechanics method. *Tetrahedron Letters*, 27(16):1825–1828, 1986.

[35] G. M. Smith and D. F. Veber. Computer-aided, systematic search of peptide conformations constrained by nmr data. *Biochemical and Biophysical Research Communications*, 134(2):907–914, 1986.

[36] N. Go and H. A. Scheraga. Ring closure and local conformational deformations of chain molecules. *Macromolecules*, 3(2):178–187, 1970.

[37] R. E. Bruccoleri and M. Karplus. Chain closure with bond angle variations. *Macromolecules*, 18(12):2767–2773, 1985.

[38] T. Ryhänen, F. J. Bermejo, J. Santoro, and M. Rico. Moltw: A program for conformational studies using potential functions—ii. algorithms for molecular coordinates and topology manipulation. *Computers Chemistry*, 11(1):13–18, 1987.

[39] D. N. White and D. H. Kitson. Computational conformational analysis of cyclohexaglycyl. *Journal of Molecular Graphics*, 4(2):112–116, 1986.

[40] R.A. Dammkoehler, S.F. Karasek, E.B. Shands, and G.R. Marshall. Constrained search of conformational hyperspace. *Journal of computer-aided molecular design*, 3:3–21, 1989.

[41] M. Lipton and W.C. Still. The multiple minimum problem in molecular modeling. tree searching internal coordinate conformational space. *Journal of Computational Chemistry*, 9(4):343–355, 1988.

[42] R.E. Bruccoleri and M. Karplus. Prediction of the folding of short polypeptide segments by uniform conformational sampling. *Biopolymers: Original Research on Biomolecules*, 26(1):137–168, 1987.

[43] B.R. Brooks, R.E. Bruccoleri, B.D. Olafson, D.J. States, S. Swaminathan, and M. Karplus. Charmm: a program for macromolecular energy, minimization, and dynamics calculations. *J. comput. Chem*, 4(2):187–217, 1983.

[44] J.M. Goodman and W.C. Still. An unbounded systematic search of conformational space. *Journal of computational chemistry*, 12(9):1110–1117, 1991.

[45] D.D. Beusen, E.B. Shands, S.F. Karasek, G.R. Marshall, and R.A. Dammkoehler. Systematic search in conformational analysis. *Journal of Molecular Structure: THEOCHEM*, 370(2-3):157–171, 1996.

[46] G.M. Crippen and H.A. Scheraga. Minimization of polypeptide energy: Xi. the method of gentlest ascent. *Archives of biochemistry and biophysics*, 144(2):462–466, 1971.

[47] I. Kolossváry and W.C. Guida. Low-mode conformational search elucidated: Application to c39h80 and flexible docking of 9-deazaguanine inhibitors into pnp. *Journal of Computational Chemistry*, 20(15):1671–1684, 1999.

[48] C. Levinthal. How to fold graciously. In J.T.P. DeBrunner and E. Munck, editors, *Mossbacher Colloquium: Conformation of Biological Molecules and Polymers*, pages 22–24, Berlin, 1969. Springer Verlag.

[49] P. Labute. Lowmodemd-implicit low-mode velocity filtering applied to conformational search of macrocycles and protein loops. *Journal of chemical information and modeling*, 50(5):792–800, 2010.

[50] C. Parish, R. Lombardi, K. Sinclair, E. Smith, A. Goldberg, M. Rappleye, and M. Dure. A comparison of the low mode and monte carlo conformational search methods. *Journal of Molecular Graphics and Modelling*, 21(2):129–150, 2002.

[51] S. Stepanenko and B. Engels. Tabu search based strategies for conformational search. *The Journal of Physical Chemistry A*, 113(43):11699–11705, 2009.

[52] S. Stepanenko and B. Engels. Gradient tabu search. *Journal of Computational Chemistry*, 28(2):601–611, 2007.

[53] C.C. Roberts and C.E.A. Chang. Ligand binding pathway elucidation for cryptophane host–guest complexes. *Journal of Chemical Theory and Computation*, 9(4):2010–2019, 2013.

[54] Y.H. Sanejouand. Normal-mode driven exploration of protein domain motions. *Journal of Computational Chemistry*, 42(31):2250–2257, 2021.

[55] S. Mahajan and Y.H. Sanejouand. Jumping between protein conformers using normal modes. *Journal of Computational Chemistry*, 38(18):1622–1630, 2017.

[56] J. Rodriguez, L. Carcache, and K.S. Rein. Low-mode docking search in iglur homology models implicates three residues in the control of ligand selectivity. *Journal of Molecular Recognition: An Interdisciplinary Journal*, 18(2):183–189, 2005.

[57] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[58] S.H. Northrup and J.A. McCammon. Simulation methods for protein structure fluctuations. *Biopolymers: Original Research on Biomolecules*, 19(5):1001–1016, 1980.

[59] T. Noguti and N. Gō. Efficient monte carlo method for simulation of fluctuating conformations of native proteins. *Biopolymers: Original Research on Biomolecules*, 24(3):527–546, 1985.

[60] Z. Li and H.A. Scheraga. Monte carlo-minimization approach to the multiple-minima problem in protein folding. *Proceedings of the National Academy of Sciences*, 84(19):6611–6615, 1987.

[61] G. Chang, W.C. Guida, and W.C. Still. An internal-coordinate monte carlo method for searching conformational space. *Journal of the American Chemical Society*, 111(12):4379–4386, 1989.

[62] M. Saunders. Stochastic exploration of molecular mechanics energy surfaces. hunting for the global minimum. *Journal of the American Chemical Society*, 109(10):3150–3152, 1987.

[63] M. Saunders. Stochastic search for the conformations of bicyclic hydrocarbons. *Journal of computational chemistry*, 10(2):203–208, 1989.

[64] M. Saunders and R.M. Jarret. A new method for molecular mechanics. *Journal of computational chemistry*, 7(4):578–588, 1986.

[65] N.L. Allinger. Conformational analysis. 130. mm2. a hydrocarbon force field utilizing v1 and v2 torsional terms. *Journal of the American Chemical Society*, 99(25):8127–8134, 1977.

[66] D.M. Ferguson and D.J. Raber. A new approach to probing conformational space with molecular mechanics: random incremental pulse search. *Journal of the American Chemical Society*, 111(12):4371–4378, 1989.

[67] H. Senderowitz, F. Guarnieri, and W.C. Still. A smart monte carlo technique for free energy simulations of multiconformational molecules. direct calculations of the conformational populations of organic molecules. *Journal of the American Chemical Society*, 117(31):8211–8219, 1995.

[68] H. Senderowitz and W.C. Still. Mc(jbw): Simple but smart monte carlo algorithm for free energy simulations of multiconformational molecules. *Journal of computational chemistry*, 19(15):1736–1745, 1998.

[69] S.R. Wilson, W. Cui, J.W. Moskowitz, and K.E. Schmidt. Conformational analysis of flexible molecules: Location of the global minimum energy conformation by the simulated annealing method. *Tetrahedron Letters*, 29(35):4373–4376, 1988.

[70] J.H. Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

[71] N. Nair and J.M. Goodman. Genetic algorithms in conformational analysis. *Journal of Chemical Information and Computer Sciences*, 38(2):317–320, 1998.

[72] M.J. Blommers, C.B. Lucasius, G. Kateman, and R. Kaptein. Conformational analysis of a dinucleotide photodimer with the aid of the genetic algorithm. *Biopolymers: Original Research on Biomolecules*, 32(1):45–52, 1992.

[73] D.B. McGarrah and R.S. Judson. Analysis of the genetic algorithm method of molecular conformation determination. *Journal of computational chemistry*, 14(11):1385–1395, 1993.

[74] R.S. Judson, E.P. Jaeger, A.M. Treasurywala, and M.L. Peterson. Conformational searching methods for small molecules. ii. genetic algorithm approach. *Journal of Computational Chemistry*, 14(11):1407–1414, 1993.

[75] R.S. Judson, M. Colvin, J.C. Meza, A. Huffer, and D. Gutierrez. Do intelligent configuration search techniques outperform random search for large molecules? *International Journal of Quantum Chemistry*, 44(2):277–290, 1992.

[76] R. Unger and J. Moult. Genetic algorithms for protein folding simulations. *Journal of molecular biology*, 231(1):75–81, 1993.

[77] H. Mühlenbein, M. Schomisch, and J. Born. The parallel genetic algorithm as function optimizer. *Parallel computing*, 17(6-7):619–632, 1991.

[78] J.A. McCammon and M. Karplus. The dynamic picture of protein structure. *Accounts of Chemical Research*, 16(6):187–193, 1983.

[79] D. Frenkel and B. Smit. *Understanding molecular simulation: from algorithms to applications*. Elsevier, 2023.

[80] R.W. Zwanzig. High-temperature equation of state by a perturbation method. i. nonpolar gases. *The Journal of Chemical Physics*, 22(8):1420–1426, 1954. SHAKE.

[81] P.H. Hünenberger. Thermostat algorithms for molecular dynamics simulations. *Advanced computer simulation: Approaches for soft matter sciences I*, pages 105–149, 2005. NVT.

[82] G.M. Torrie and J.P. Valleau. Nonphysical sampling distributions in monte carlo free-energy estimation: Umbrella sampling. *Journal of Computational Physics*, 23(2):187–199, 1977. Umbrella sampling.

[83] J.G. Kirkwood. Statistical mechanics of fluid mixtures. *The Journal of chemical physics*, 3(5):300–313, 1935. TI.

[84] R.W. Zwanzig. High-temperature equation of state by a perturbation method. i. nonpolar gases. *The Journal of Chemical Physics*, 22(8):1420–1426, 1954.

[85] A. Laio and M. Parrinello. Escaping free-energy minima. *Proceedings of the national academy of sciences*, 99(20):12562–12566, 2002. Meta.

[86] G.M. Crippen and T.F. Havel. *Distance geometry and molecular conformation*, volume 74. Research Studies Press, Taunton, 1988.

[87] G. Landrum. Rdkit: A software suite for cheminformatics, computational chemistry, and predictive modeling, 2013. Greg Landrum, 8, p.31.

[88] R.V.N. Melnik, A. Uhlherr, J. Hodgkin, and F. De Hoog. Distance geometry algorithms in molecular modelling of polymer and composite systems. *Computers & Mathematics with Applications*, 45(1-3):515–534, 2003.

[89] J. P. Ebejer, G. M. Morris, and C. M. Deane. Freely available conformer generation methods: how good are they? *Journal of Chemical Information and Modeling*, 52(5):1146–1158, 2012.

[90] S. Riniker and G.A. Landrum. Better informed distance geometry: Using what we know to improve conformation generation. *Journal of Chemical Information and Modeling*, 55(12):2562–2574, 2015.

[91] C.R. Groom, I.J. Bruno, M.P. Lightfoot, and S.C. Ward. The cambridge structural database. *Acta Crystallographica Section B: Structural Science, Crystal Engineering and Materials*, 72(2):171–179, 2016.

[92] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, and P.E. Bourne. The protein data bank. *Nucleic acids research*, 28(1):235–242, 2000.

[93] J. Gasteiger, C. Rudolph, and J. Sadowski. Automatic generation of 3d-atomic coordinates for organic molecules. *Tetrahedron Computer Methodology*, 3(6):537–547, 1990.

[94] P. C. D. Hawkins, A. G. Skillman, G. L. Warren, B. A. Ellingson, and M. T. Stahl. Conformer generation with omega: algorithm and validation using high quality structures from the protein databank and cambridge structural database. *Journal of chemical information and modeling*, 50(4):572–584, 2010.

[95] N.M. O'Boyle, T. Vandermeersch, C.J. Flynn, A.R. Maguire, and G.R. Hutchison. Confab-systematic generation of diverse low-energy conformers. *Journal of cheminformatics*, 3(1):1–9, 2011.

[96] K.S. Watts, P. Dalal, R.B. Murphy, W. Sherman, R.A. Friesner, and J.C. Shelley. Confgen: a conformational search method for efficient generation of bioactive conformers. *Journal of chemical information and modeling*, 50(4):534–546, 2010.

[97] J. Li, T. Ehlers, J. Sutter, S. Varma-O'Brien, and J. Kirchmair. Caesar: a new conformer generation algorithm based on recursive buildup and local rotational symmetry consideration. *Journal of chemical information and modeling*, 47(5):1923–1932, 2007.

[98] M.A. Miteva, F. Guyon, and P. Tufféry. Frog2: Efficient 3d conformation ensemble generator for small compounds. *Nucleic acids research*, 38(2):622–627, 2010.

[99] R.J. Loncharich, E. Seward, S.B. Ferguson, F.K. Brown, F. Diederich, and K.N. Houk. On the origin of cavity-filling conformations of macrocycles: a 1h nmr spectroscopic and force-field computational study. *The Journal of Organic Chemistry*, 53(15):3479–3491, 1988.

[100] N.Z. Shor. Cut-off method with space extension in convex programming problems. *Cybernetics*, 13(1):94–96, 1977.

[101] M. Billeter, T.F. Havel, and K. Wüthrich. The ellipsoid algorithm as a method for the determination of polypeptide conformations from experimental distance constraints and energy minimization. *Journal of computational chemistry*, 8(2):132–141, 1987.

[102] D.P. Dolata, A.R. Leach, and K. Prout. Wizard: Ai in conformational analysis. *Journal of Computer-Aided Molecular Design*, 1:73–85, 1987.

[103] L. Chan, G.R. Hutchison, and G.M. Morris. Bayesian optimization for conformer generation. *Journal of cheminformatics*, 11(1):1–11, 2019.

[104] L. Chan, G.R. Hutchison, and G.M. Morris. Bokei: Bayesian optimization using knowledge of correlated torsions and expected improvement for conformer generation. *Physical Chemistry Chemical Physics*, 22(9):5211–5219, 2020.

[105] E. Mansimov, O. Mahmood, S. Kang, and K. Cho. Molecular geometry prediction using a deep generative graph neural network. *Scientific reports*, 9(1):20381, 2019.

[106] M. Xu, L. Yu, Y. Song, C. Shi, S. Ermon, and J. Tang. Geodiff: A geometric diffusion model for molecular conformation generation. *arXiv preprint arXiv:2203.02923*, 2022.

[107] G.N. Simm and J.M. Hernández-Lobato. A generative model for molecular distance geometry. *arXiv preprint arXiv:1909.11459*, 2019.

[108] K. Schütt, P.J. Kindermans, H.E. Sauceda Felix, S. Chmiela, A. Tkatchenko, and K.R. Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. In *Advances in neural information processing systems*, volume 30, 2017.

[109] M. Xu, S. Luo, Y. Bengio, J. Peng, and J. Tang. Learning neural generative dynamics for molecular conformation generation. *arXiv preprint arXiv:2102.10240*, 2021.

[110] O. Ganea, L. Pattanaik, C. Coley, R. Barzilay, K. Jensen, W. Green, and T. Jaakkola. Geomol: Torsional geometric generation of molecular 3d conformer ensembles. In *Advances in Neural Information Processing Systems*, volume 34, pages 13757–13769, 2021.

[111] M. Xu, W. Wang, S. Luo, C. Shi, Y. Bengio, R. Gomez-Bombarelli, and J. Tang. An end-to-end framework for molecular conformation generation via bilevel programming. In *International Conference on Machine Learning*, pages 11537–11547. PMLR, 2021.

[112] C. Shi, S. Luo, M. Xu, and J. Tang. Learning gradient fields for molecular conformation generation. In *International conference on machine learning*, pages 9558–9568. PMLR, 2021.

[113] J. Han, L. Zhang, and R. Car. Deep potential: A general representation of a many-body potential energy surface. *arXiv preprint arXiv:1707.01478*, 2017.

[114] H. Wang, L. Zhang, J. Han, and E. Weinan. Deepmd-kit: A deep learning package for many-body potential energy representation and molecular dynamics. *Computer Physics Communications*, 228:178–184, 2018.

[115] L. Zhang, J. Han, H. Wang, R. Car, and E.J.P.R.L. Weinan. Deep potential molecular dynamics: a scalable model with the accuracy of quantum mechanics. *Physical review letters*, 120(14):143001, 2018.

[116] Y. Zhang, H. Wang, W. Chen, J. Zeng, L. Zhang, H. Wang, and E. Weinan. Dp-gen: A concurrent learning platform for the generation of reliable deep learning based potential energy models. *Computer Physics Communications*, 253:107206, 2020.

[117] N.L. Allinger, Y.H. Yuh, and J.H. Lii. Molecular mechanics. the mm3 force field for hydrocarbons. 1. *Journal of the American Chemical Society*, 111(23):8551–8566, 1989.

[118] J.W. Ponder and D.A. Case. Force fields for protein simulations. *Advances in protein chemistry*, 66:27–85, 2003.

[119] B.R. Brooks, R.E. Bruccoleri, B.D. Olafson, D.J. States, S.A. Swaminathan, and M. Karplus. Charmm: a program for macromolecular energy, minimization, and dynamics calculations. *Journal of computational chemistry*, 4(2):187–217, 1983.

[120] T.A. Halgren. Merck molecular force field. i. basis, form, scope, parameterization, and performance of mmff94. *Journal of computational chemistry*, 17(5-6):490–519, 1996.

[121] W.L. Jorgensen, D.S. Maxwell, and J. Tirado-Rives. Development and testing of the opls all-atom force field on conformational energetics and properties of organic liquids. *Journal of the American Chemical Society*, 118(45):11225–11236, 1996.

[122] N.O. Friedrich, C. de Bruyn Kops, F. Flachsenberg, K. Sommer, M. Rarey, and J. Kirchmair. Benchmarking commercial conformer ensemble generators. *Journal of chemical information and modeling*, 57(11):2719–2728, 2017.

[123] M. Saunders, K.N. Houk, Y.D. Wu, W.C. Still, M. Lipton, G. Chang, and W.C. Guida. Conformations of cycloheptadecane. a comparison of methods for conformational searching. *Journal of the American Chemical Society*, 112(4):1419–1427, 1990.

[124] J.M. Goodman. What is the longest unbranched alkane with a linear global minimum conformation? *Journal of chemical information and computer sciences*, 37(5):876–878, 1997.

[125] C. Lu, C. Wu, D. Ghoreishi, W. Chen, L. Wang, W. Damm, G.A. Ross, M.K. Dahlgren, E. Russell, C.D. Von Bargen, et al. Opls4: Improving force field accuracy on challenging regimes of chemical space. *Journal of chemical theory and computation*, 17(7):4291–4300, 2021.

[126] E. Polak and G. Ribiere. Note sur la convergence de methodes de directions conjuguees. *Revue Francaise Inf. Rech. Oper., Serie Rouge*, 16-R1:35, 1969.

[127] R.S. Paton and J.M. Goodman. Exploration of the accessible chemical space of acyclic alkanes. *Journal of chemical information and modeling*, 47(6):2124–2132, 2007.

[128] S. Axelrod and R. Gomez-Bombarelli. Geom, energy-annotated molecular conformations for property prediction and molecular generation. *Scientific Data*, 9(1):185, 2022.

[129] J. J. Irwin and B. K. Shoichet. Zinc- a free database of commercially available compounds for virtual screening. *Journal of chemical information and modeling*, 45(1):177–182, 2005.

[130] A. Gaulton, L. J. Bellis, A. P. Bento, J. Chambers, M. Davies, A. Hersey, Y. Light, S. McGlinchey, D. Michalovich, B. Al-Lazikani, et al. Chembl: a large-scale bioactivity database for drug discovery. *Nucleic acids research*, 40(D1):D1100–D1107, 2012.

[131] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[132] B. E Suzek, H. Huang, P. McGarvey, R. Mazumder, and C. H. Wu. Uniref: comprehensive and non-redundant uniprot reference clusters. *Bioinformatics*, 23(10):1282–1288, 2007.

Appendix A

These projects' scripts and original data are available from the below provided links

Part 1 : *Diamond Energy* Method
https://github.com/Goodman-lab/Diamond_energy

Part 2 : Further Development of *Diamond Energy*
https://github.com/Goodman-lab/Diamond-Energy-II

Part 3 : *Diamond Energy* Extension Work – Diamond AI
https://github.com/Goodman-lab/Diamond-Energy-III