

Master's thesis

Predicting Frictional Properties of Graphene Kirigami Using Molecular Dynamics and Neural Networks

Designs for a negative friction coefficient

Mikkel Metzsch Jensen

Computational Science: Materials Science
60 ECTS study points

Department of Physics
Faculty of Mathematics and Natural Sciences

Spring 2023



Mikkel Metzsch Jensen

Predicting Frictional Properties of Graphene Kirigami Using Molecular Dynamics and Neural Networks

Designs for a negative friction coefficient

Supervisors:
Henrik Andersen Sveinsson
Anders Malthe-Sørensen

Abstract

Various theoretical models and experimental results propose different governing mechanisms for friction at the nanoscale. We consider a graphene sheet modified with Kirigami-inspired cuts and under the influence of strain. Prior research has demonstrated that this system exhibits out-of-plane buckling, which could result in a decrease in contact area when sliding on a substrate. According to asperity theory, this decrease in contact area is expected to lead to a reduction of friction. However, to the best of our knowledge, no previous studies have investigated the friction behavior of a nanoscale Kirigami graphene sheet under strain. Here we show that specific Kirigami designs yield a non-linear dependency between kinetic friction and the strain of the sheet. Using molecular dynamics simulations, we have found a non-monotonic increase in friction with strain. We found that the friction-strain relationship does not show any clear dependency on contact area which contradicts asperity theory. Our findings suggest that the effect is associated with the out-of-plane buckling of the graphene sheet and we attribute this to a commensurability effect. By mimicking a load-strain coupling through tension, we were able to utilize this effect to demonstrate a negative friction coefficient on the order of -0.3 for loads in the range of a few nN. In addition, we have attempted to use machine learning to capture the relationship between Kirigami designs, load, and strain, with the objective of performing an accelerated search for new designs. Although this approach yielded some promising results, we conclude that further improvements to the dataset are necessary in order to develop a reliable model. We anticipate our findings to be a starting point for further investigations of the underlying mechanism for the frictional behavior of a Kirigami sheet. For instance, the commensurability hypothesis could be examined by varying the sliding angle in simulations. We propose to use an active learning strategy to extend the dataset for the use of machine learning to assist these investigations. If successful, further studies can be done on the method of inverse design. In summary, our findings suggest that the application of nanoscale Kirigami can be promising for developing novel friction-control strategies.

Acknowledgments

The task of writing a master's thesis is a demanding and extensive project which I could not have done without the support of many good people around me. First of all, I want to thank my supervisors Henrik Andersen Sveinsson and Anders Malthe-Sørensen for the assistance in this thesis work. I am especially grateful for the weekly meetings with Henrik and the inspiring discussions had as we unraveled the discoveries related to the topic of this thesis. I remember that I initially asked for an estimate of how much time he had available for supervision and the answer was something along the lines of "There are no limits really, just send me an email and we figure it out". This attitude captures the main experience I have had working with Henrik and I am profoundly grateful for the time and effort he has put into this project. I hope that he did not regret this initial statement too much, because I have certainly been taken advantage of it. I also want to thank Even Marius Nordhagen for technical support regarding the use of the computational cluster. In that context, I also want to acknowledge the Center for Computing in Science Education (CCSE) for making these resources available.

I would like to express my gratitude to all the parties involved in making it possible for me to write my thesis from Italy. I am particularly grateful for the flexibility shown by my supervisors and for the support of Anders Kvellestad, who allowed me to work remotely as a group teacher. I would also like to thank Scuola Normale Superiore for providing me with access to their library.

I realize that it is a commonly used cliché to express gratitude for the support of loved ones. However, I want to highlight the exceptional role played by my fiancé, Ida, who deserves the main credit for enabling me to maintain a healthy state of mind. She has provided me with a solid foundation for a fulfilling life that enables me to pursue secondary objectives, such as an academic career. I look forward to spending the rest of my life with you.

In this thesis, I have used the formal pronoun "we" mainly as a customary habit related to the formalities of scientific writing in a team. Nonetheless, I have realized that this usage is more fitting as I have not been working alone on this project. I have received support all the way from colleagues and friends at the University of Oslo, my family residing in Denmark, and my life partner who slept beside me every night here in Italy. They are the "good people around me" who have made this thesis possible.

Acronyms

AFM Atomic Force Microscope.

CM Center of Mass.

CNN Convolutional Neural Network.

EMA Exponetial Moving Average.

FFM Friction Force Microscopy.

FK Frenkel-Kontorova.

FKT Frenkel-Kontorova–Tomlinson.

GA Genetic Algorithm.

GAN Generative Adversarial Networks.

GS Ground State.

LJ Lennard-Jones.

MD Molecular Dynamics.

ML Machine Learning.

MSE Mean Squared Error.

NN Nearest Neighbors.

PT Prandtl–Tomlinson.

RMSProp Root Mean Square Propagation.

SFA Surface Force Apparatus.

SFM Scanning Force Microscopy.

SGD Stochastic Gradient Descent.

SPM Scanning Probe Microscopy.

std Standard Deviation.

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 1 |
| 1.1 | Motivation | 1 |
| 1.2 | Goals | 2 |
| 1.3 | Contributions | 3 |
| 1.4 | Thesis structure | 3 |
| I | Background Theory | 5 |
| 2 | Friction | 7 |
| 2.1 | Friction across scales | 7 |
| 2.2 | Macroscale | 7 |
| 2.2.1 | Amontons' law | 7 |
| 2.3 | Microscopic scale | 9 |
| 2.3.1 | Asperity theories | 9 |
| 2.4 | Nanoscale — Atomic scale | 10 |
| 2.4.1 | Prandtl-Tomlinson | 11 |
| 2.4.1.1 | Thermal activation | 12 |
| 2.4.1.2 | Sliding speed | 13 |
| 2.4.1.3 | Tip mass | 13 |
| 2.4.1.4 | Friction regimes: Smooth sliding, single slip, and multiple slip | 13 |
| 2.4.2 | Frenkel-Kontorova | 15 |
| 2.4.2.1 | Commensurability | 15 |
| 2.4.2.2 | Velocity resonance | 17 |
| 2.4.3 | Frenkel-Kontorova-Tomlinson | 18 |
| 2.4.4 | Shortcomings of the reduced-models | 18 |
| 2.4.5 | Experimental procedures | 19 |
| 2.4.5.1 | Scanning Probe Microscopy | 19 |
| 2.4.5.2 | Surface Force Apparatus | 20 |
| 2.5 | Summary of previous results | 20 |
| 2.6 | Research questions | 23 |
| 3 | Molecular Dynamics | 25 |
| 3.1 | Potentials | 25 |
| 3.1.1 | General formulation of potentials | 25 |
| 3.1.2 | Lennard Jones | 26 |
| 3.1.3 | Stillinger-Weber | 27 |
| 3.1.4 | Tersoff | 28 |
| 3.2 | Integration | 30 |
| 3.2.1 | Velocity Verlet | 30 |
| 3.3 | Thermostats | 31 |
| 3.3.1 | Langevin thermostat | 31 |
| 3.3.2 | Applying the Langevin Thermostat | 33 |
| 3.4 | Limitations | 33 |
| 3.5 | LAMMPS | 34 |

| | |
|---|-----------|
| 4 Machine Learning | 35 |
| 4.1 Neural network | 35 |
| 4.1.1 Optimizers | 37 |
| 4.1.2 Weight decay | 38 |
| 4.1.3 Parameter distributions | 39 |
| 4.1.4 Learning rate decay strategies | 39 |
| 4.2 Convolutional Neural Network | 40 |
| 4.2.1 Training, validation and test data | 42 |
| 4.3 Overfitting and underfitting | 42 |
| 4.4 Hypertuning | 43 |
| 4.5 Prediction explanation | 45 |
| 4.6 Accelerated search using genetic algorithm | 45 |
| II Simulations | 49 |
| 5 Creating a graphene kirigami system | 51 |
| 5.1 Region definitions | 52 |
| 5.2 Numerical procedure | 54 |
| 5.3 Setting up the substrate | 54 |
| 5.4 Setting up sheet | 55 |
| 5.4.1 Indexing | 55 |
| 5.4.2 Removing atoms | 56 |
| 5.5 Kirigami patterns | 58 |
| 5.5.1 Tetrahedron | 58 |
| 5.5.2 Honeycomb | 60 |
| 5.5.3 Random walk | 60 |
| 5.5.3.1 Fundamentals | 61 |
| 5.5.3.2 Spacing of walking paths | 62 |
| 5.5.3.3 Bias | 62 |
| 5.5.3.4 Stay or break | 63 |
| 5.5.3.5 Deployment schemes | 64 |
| 5.5.3.6 Validity | 65 |
| 5.5.3.7 Random walk examples | 66 |
| 6 Pilot study | 69 |
| 6.1 Friction simulation parameters | 69 |
| 6.2 Force traces | 69 |
| 6.2.1 Force oscillations | 69 |
| 6.2.2 Decompositions | 73 |
| 6.2.3 Center of mass path | 73 |
| 6.3 Defining metrics for friction | 74 |
| 6.3.1 Kinetic friction | 74 |
| 6.3.2 Static friction | 75 |
| 6.4 Out-of-plane buckling | 77 |
| 6.5 Investigating default parameters | 78 |
| 6.5.1 Computational cost | 80 |
| 6.6 Strain and load dependencies | 80 |
| 6.6.1 Pressure reference for normal load | 80 |
| 6.6.2 Strain dependency | 81 |
| 6.6.3 Load dependency | 82 |
| 6.6.4 Prospects for achieving a negative friction coefficient | 86 |
| 7 Kirigami configuration exploration | 87 |
| 7.1 Generating the dataset | 87 |
| 7.2 Data analysis | 88 |
| 7.3 Properties of interest | 88 |
| 7.4 Machine learning | 94 |
| 7.4.1 Architecture | 94 |
| 7.4.2 Data handling | 95 |

| | | |
|-------------------|---|------------|
| 7.4.2.1 | Input | 95 |
| 7.4.2.2 | Output | 95 |
| 7.4.2.3 | Data augmentation | 95 |
| 7.4.3 | Loss | 95 |
| 7.4.4 | Hypertuning | 96 |
| 7.4.5 | Model performance | 101 |
| 7.5 | Accelerated Search | 104 |
| 7.5.1 | Patteren generation search | 104 |
| 7.5.2 | Genetic algorithm search | 108 |
| 8 | Negative friction coefficient | 113 |
| 8.1 | Nanomachine design | 113 |
| 8.2 | Simulating the load-tension-coupled system | 113 |
| 9 | Summary | 117 |
| 9.1 | Summary and conclusions | 117 |
| 9.1.1 | Designing an MD simulation | 117 |
| 9.1.2 | Generating Kirigami patterns | 118 |
| 9.1.3 | Friction control using Kirigami design and strain | 118 |
| 9.1.4 | Capturing trends with machine learning | 119 |
| 9.1.5 | Accelerated search for Kirigami patterns | 120 |
| 9.1.6 | Negative friction coefficient | 120 |
| 9.2 | Outlook | 121 |
| Appendices | | 123 |
| A | Strain simulation frames | 125 |
| B | Dataset Kirigami patterns | 129 |
| B.1 | Tetrahedron | 130 |
| B.2 | Honeycomb | 133 |
| B.3 | Random walk | 135 |
| C | Friction-strain profiles | 141 |

Chapter 1

Introduction

1.1 Motivation

Friction is the force that prevents the relative motion of objects in contact. In our everyday life, we recognize it as the inherent resistance to sliding motion. Some surfaces appear slippery and some appear rough, and we know intuitively that sliding down a snow-covered hill is much more exciting than its grassy counterpart. Without friction, it would not be possible to walk across a flat surface, lean against the wall without falling over or secure an object by the use of nails or screws [1, p. 5]. It is probably safe to say that the concept of friction is integrated into our everyday life to such an extent that most people take it for granted. However, the efforts to control friction date back to the early civilization (3500 B.C.) with the use of the wheel and lubricants to reduce friction in translational motion [2]. Today, friction is considered a part of the wider field *tribology* derived from the Greek word *tribos* meaning “rubbing”. It includes the science of friction, wear and lubrication [2]. The most compelling motivation to study tribology is ultimately to gain full control of friction and wear for various technical applications. Especially, the reduction of friction is of great interest since this can be utilized to improve energy efficiency in mechanical systems with moving parts. Hence, it has been reported that tribological problems have a significant potential for both economic and environmental improvements [3]:

“On global scale, these savings would amount to 1.4% of the GDP annually and 8.7% of the total energy consumption in the long term.” [4].

On the other hand, the reduction of friction is not the only sensible application for tribological studies. Controlling frictional properties, besides minimization, might be of interest in the development of a grasping robot where finetuned object handling is required. While achieving a certain “constant” friction response is readily obtained through appropriate material choices, we are yet to unlock the full capabilities to alter friction dynamically on the go. One example from nature inspiring us to think along these lines is the gecko feet. More precisely, the Tokay gecko has received a lot of attention in scientific studies aiming to unravel the underlying mechanism of its “toggable” adhesion properties. Although the gecko can produce large adhesive forces, it retains the ability to remove its feet from an attachment surface at will [5]. This makes the gecko able to achieve a high adhesion on the feet when climbing a vertical surface while lifting them for the next step remains relatively effortless. For a grasping robot, we might consider an analog frictional concept of a surface material that can change from slippery to rough on demand depending on specific tasks; slippery and smooth when interacting with people and rough and firmly gripping when moving heavy objects.

In recent years an increasing amount of interest has gone into the studies of the microscopic origins of friction, due to the increased possibilities in surface preparation and the development of nanoscale experimental methods. Nano-friction is also of great concern for the field of nano-machining where the frictional properties between the tool and the workpiece dictate machining characteristics [3]. With concurrent progress in computational capacity and development of Molecular Dynamics (MD), numerical investigations serve as an invaluable tool for getting insight into the nanoscale mechanics associated with friction. This simulation-based approach can be considered as a “numerical experiment” enabling us to create and probe a variety of high-complexity systems which are still out of reach for modern experimental methods.

In materials science such MD-based numerical studies have been used to explore the concept of so-called *metamaterials* where the material compositions are designed meticulously to enhance certain physical properties [6–11]. This is often achieved either by intertwining different material types or removing certain regions completely. In recent papers by Hanakata et al. [6, 7], numerical studies have showcased that the mechanical properties of a graphene sheet, yield stress and yield strain, can be altered through the



introduction of so-called *Kirigami* inspired cuts into the sheet. Kirigami is a variation of origami where the paper is cut additionally to being folded. While these methods originate as an art form, aiming to produce various artistic objects, they have proven to be applicable in a wide range of fields such as optics, physics, biology, chemistry and engineering [12]. Various forms of stimuli enable direct 2D to 3D transformations through the folding, bending, and twisting of microstructures. While original human designs have contributed to specific scientific applications in the past, the future of this field is highly driven by the question of how to generate new designs optimized for certain physical properties. However, the complexity of such systems and the associated design space makes for seemingly intractable¹ problems ruling out analytic solutions.

Earlier design approaches such as bioinspiration, looking at gecko feet for instance, and Edisonian, based on trial and error, generally rely on prior knowledge and an experienced designer [9]. While the Edisonian approach is certainly more feasible through numerical studies than real-world experiments, the number of combinations in the design space rather quickly becomes too large for a systematic search, even when considering the computation time on modern-day hardware. However, this computational time constraint can be relaxed by the use of machine learning (ML) which has been proven successful in the establishment of a mapping from the design space to physical properties of interest. This gives rise to two new styles of design approaches: One, by utilizing the prediction from a trained network we can skip the MD simulations altogether resulting in an *accelerated search* of designs. This can be further improved by guiding the search according to the most promising candidates. For instance, as done with the *genetic algorithm* based on mutation and crossing. Another more sophisticated approach is through generative methods such as *Generative Adversarial Networks* (GAN) or diffusion models. The latter is being used in state-of-the-art AI systems such as OpenAI's DALL-E2 [13] or Midjourney [14]. By working with a so-called *encoder-decoder* network structure, one can build a model that reverses the prediction process. This is often referred to as *inverse design*, where the model predicts a design based on physical target properties. In the papers by Hanakata et al. [6, 7] both the accelerated search and the inverse design approach was proven successful to create novel metamaterial Kirigami designs with the graphene sheet.

Hanakata et al. attribute the variation in mechanical properties to the non-linear effects arising from the out-of-plane buckling of the sheet. Since it is generally accepted that the surface roughness is of great importance for frictional properties it can be hypothesized that Kirigami-induced out-of-plane buckling can also be exploited for the design of frictional metamaterials. For certain designs, we might hope to find a relationship between the stretching of the sheet and frictional properties. If significant, this could give rise to an adjustable friction beyond the point of manufacturing. For instance, the grasping robot might apply such a material as artificial skin for which stretching or relaxing of the surface could result in a changeable friction strength.

In addition, the Kirigami graphene properties can be explored through a potential coupling between the strain and the normal load, through a nanomachine design, with the aim of altering the friction coefficient. This invites the idea of non-linear friction coefficients which might in principle also take on negative values. This would constitute a rarely found property which is mainly observed for the unloading phase of adhesive surfaces [15] or in the loading phase of particular heterojunction materials [16, 17].

To the best of our knowledge, Kirigami has not yet been implemented to alter the frictional properties of a nanoscale system. However, in a recent paper by Liefferink et al. [18] it is reported that macroscale Kirigami can be used to dynamically control the macroscale roughness of a surface through stretching. They reported that the roughness change led to a changeable frictional coefficient by more than one order of magnitude. This supports the idea that Kirigami designs can be used to alter friction, but we believe that taking this concept to the nanoscale would involve a different set of governing mechanisms and thus contribute to new insight in this field.

1.2 Goals

In this thesis, we investigate the prospects of altering the frictional properties of a graphene sheet through the application of Kirigami-inspired cuts and stretching of the sheet. With the use of molecular dynamics (MD) simulations, we evaluate the frictional properties of various Kirigami designs under different physical conditions. Based on the MD results, we investigate the possibility to use machine learning for the prediction of frictional properties and subsequently using the model for an accelerated search of new designs. The main goals of the thesis can be summarized as follows.

¹In computer science we define an *intractable* problem as a problem with no *efficient* algorithm to solve it nor any analytical solutions. The only way to solve such problems is the *brute-force* approach, simply trying all possible combinations, which is often beyond the capabilities of computational resources.

1. Design an MD simulation procedure to evaluate the frictional properties of a Kirigami graphene sheet under specified physical conditions.
2. Develop a numerical tool to generate various Kirigami designs, both by seeking inspiration from macroscale designs and by the use of a random-walk-based algorithm.
3. Investigate the frictional behavior under varying strain and load for different Kirigami designs.
4. Develop and train a machine learning model to predict the MD simulation results and perform an accelerated search of new designs with the goal of optimizing certain frictional properties.

1.3 Contributions

By working towards the goals outlined above (Sec. 1.2), I have discovered a non-linear relationship between the kinetic friction and the strain for certain Kirigami patterns. This phenomenon was found to be associated with the out-of-plane buckling of the Kirigami sheet but with no clear relationship to the contact area or the tension in the sheet. I found that this method does not provide any mechanism for a reduction in friction, in comparison to a non-cut sheet. However, the straining of certain Kirigami sheets allows for a non-monotonic increase in friction. The relationship to normal load was proven negligible in this context and I have demonstrated that a coupled system of load and strain (through sheet tension) can exhibit a negative friction coefficient in certain load ranges. Moreover, I have created a dataset of roughly 10,000 data points for assessing the employment of machine learning and accelerated search of Kirigami designs. I have found, that this approach might be useful, but that it requires an extended dataset in order to produce reliable results for a search of new designs.

During my investigations, I have built three numerical tools, in addition to the usual scripts for data analysis, which are available on Github [19]. The tools are summarized in the following.

- I have written a LAMMPS-based [20] tool for simulating and measuring the frictional properties of a graphene sheet sliding on a substrate. The code is generally made flexible with regard to the choice of sheet configuration, system size, simulation parameters and MD potentials, which makes it applicable for further studies on this topic. I have also built an automated procedure to carry out multiple simulations under varying parameters by submitting jobs to a computational cluster via an ssh connection. This was done by adding minor additions to the Python package developed by E. M. Nordhagen [21].
- I have generated a Python-based tool for generating Kirigami patterns and exporting these in a compatible format with the simulation software created. The generation of molecular structures is done with the use of ASE [22]. Our software includes two classes of patterns inspired by macroscale designs and a random walk algorithm which allows for a variety of different designs through user-defined biases and constraints. Given our system size of choice, the first two pattern generators are capable of generating on the order of 10^8 unique designs while the random walk generator allows for significantly more.
- I have built a machine-learning tool based on Pytorch [23] which includes setting up the data loaders, a convolutional network architecture, a loss function, and general algorithms for training and validating the results. Additionally, I have written several scripts for performing grid searches and analyzing the model predictions in the context of the frictional properties of graphene.

All numerical implementations have been originally developed for this thesis except for the libraries mentioned above along with common Python libraries such as Numpy and Matplotlib.

1.4 Thesis structure

The thesis is divided into two parts. In Part I we introduce the relevant theoretical background, and in Part II we present the numerical implementations and the results of this thesis.

Part I contains a description of the theoretical background related to Friction (Chapter 2), Molecular Dynamics (Chapter 3) and Machine Learning (Chapter 4). In Sec. 2.6 we formulate our research questions in the light of the friction theory.

In Part II, we begin by presenting the system in Chapter 5 which includes a definition of the main parts of the system and the numerical procedures related to the MD simulation. Here we also present the generation of Kirigami designs. In Chapter 6 we carry out a pilot study where we evaluate the simulation results for



various physical conditions and compare a non-cut sheet to two different Kirigami designs. In Chapter 7, we further explore the Kirigami patterns through the creation of a dataset and the employment of machine learning and an accelerated search for new designs. In Chapter 8, we use the results from the pilot study to demonstrate the possibility to achieve a negative friction coefficient for a system with coupled load and strain. Finally, in Chapter 9, we summarize our results and provide an outlook for further studies. Additional figures are shown in Appendix A, Appendix B and Appendix C.

Part I

Background Theory



Chapter 2

Friction

Since we aim for controlling frictional properties, we will review the relevant theoretical understanding of friction in this chapter. We limit ourselves to the tribological subcategory, wear-less dry friction, meaning that we consider friction in the absence of any lubricant and wear between the contacting surfaces. We will direct the review towards our system of interest consisting of a nanoscale graphene sheet sliding on a substrate. This will serve as a basis for a formal definition of our research questions at the end of this chapter.

2.1 Friction across scales

Tribological systems span a wide range of time and length scales, from geological stratum layers involved in earthquakes [3] to atomistic processes, such as the gliding motion of nanoclusters or nanomotors [24]. This vast difference in scale leads to different dominant frictional mechanisms. At the macroscale, the experimental systems are typically subjected to relatively high loads and sliding speeds, resulting in significant contact stress and wear. This makes for a macroscale friction that is often reduced into a few variables such as load, material type, sliding speed and surface roughness. On the other hand, the micro-/nanoscale regime is usually studied in the opposite domain operating under a relatively small load and sliding speed with negligible wear [3] [2, p. 5]. This reveals a change in the dominant mechanism at play with an emphasis on the importance of surface properties. The work of Bhushan and Kulkarni [25] showed that the friction coefficient decreased with scale even though the materials used were unchanged. This reveals an intrinsic relationship between friction and scale as the contact condition is altered. The phenomenological descriptions of macroscale friction cannot yet be derived from the fundamental atomic principles, and bridging the gap between different length scales in tribological systems remains an open challenge [24]. Hence, the following sections will be organized into macroscale (Sec. 2.2), microscale (Sec. 2.3) and nanoscale (Sec. 2.4) representing the theoretical understanding governing each scale regime. Realizing that the field of friction across all scales is a vastly broad topic, we will only introduce the most essential findings for each scale while keeping a main focus on features associated with our system at the nanoscale.

2.2 Macroscale

Our working definition of the *macroscale* is everything on the scale of millimeters and above [26]. This represents the scale of visible objects and includes items from our everyday life to big geological systems.

2.2.1 Amontons' law

In order to start and keep a solid block moving against a solid surface we must overcome certain frictional forces F_{fric} [1]. The static friction force F_s corresponds to the minimum tangential force required to initiate the sliding while the kinetic friction force F_k corresponds to the tangential force needed to sustain such a sliding at a steady speed. The work of Leonardo da Vinci (1452–1519), Guillaume Amontons (1663–1705) and Charles de Coulomb (1736–1806) all contributed to the empirical law, commonly known as *Amontons' law*, which serves as a common base for macroscale friction. Amontons' law states that the frictional forces are entirely independent of contact area and sliding velocity. Instead, it relies only on the normal force F_N , acting perpendicular to the surface, and the material-specific friction coefficient μ as

$$F_{\text{fric}} = \mu F_N. \quad (2.1)$$



Notice that the term *normal force* is often used interchangeably with *load* and *normal load* although the load and normal load refer to the applied force that pushes the object into the surface, whereas the normal force is the reaction force acting from the surface on the object. In equilibrium, these forces are equal in magnitude but opposite in direction. We will mainly consider a system that is in equilibrium with respect to the loading direction, and thus we will refer to the magnitude of the forces and will not distinguish between these terms either. On the same note, we point out that the friction force is different from a conventional force which in the Newtonian definition acts on a body from the outside and makes it accelerate [27]. Rather than being an independent external force the friction force is an internal *reaction* force opposing the externally applied “sliding” force.

The friction coefficient μ is typically different for the cases of static (μ_s) and kinetic (μ_k) friction, usually both with values lower than one and $\mu_s \geq \mu_k$ in all cases [1, p. 6]. The friction coefficient is taken to be a constant defined by either [27]

$$\mu_1 = \frac{F_{\text{fric}}}{F_N}, \quad (2.2a) \quad \text{or} \quad \mu_2 = \frac{dF_{\text{fric}}}{dF_N}. \quad (2.2b)$$

The first definition Eq. (2.2a) requires zero friction at zero load, i.e. $F_{\text{fric}} = 0$ at $F_N = 0$, while the second definition Eq. (2.2b) allows for a finite friction force at zero load as the coefficient is defined by the slope of the friction-load curve. The consequences of these definitions are illustrated in Fig. 2.1, for selected friction-load-curves in Fig. 2.1a and corresponding friction coefficients in Fig. 2.1b and Fig. 2.1c. For adhesive contacts, the friction force will not be zero under zero load [27] (red curve: Linear + shift) which can be mitigated by adding an extra constant to Amontons’ law (Eq. (2.1)). Using Eq. (2.2a) for adhesive contacts would make the friction coefficient diverge for decreasing load as illustrated in Fig. 2.1b. Thus, we find the second definition Eq. (2.2b) more robust and versatile. This also allows for a better interpretation of the friction coefficient in the case where friction depends non-linearly on load as seen with the purple curve in Fig. 2.1.

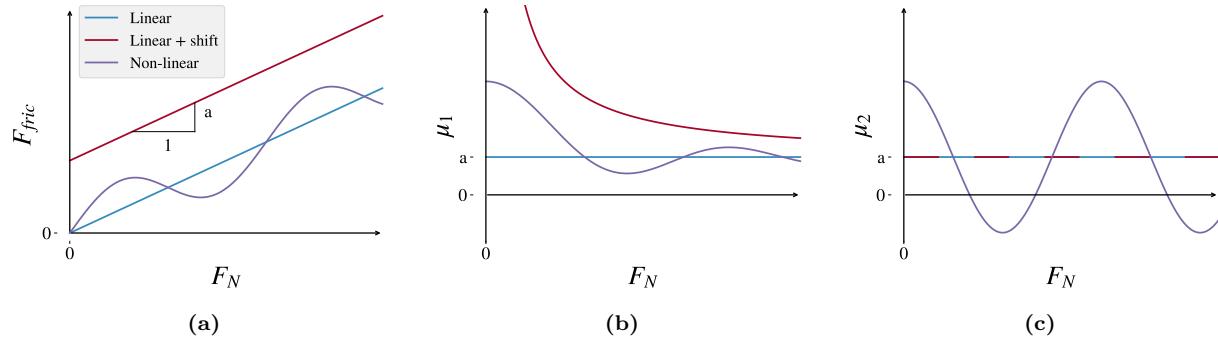


Figure 2.1: Illustration of the consequences for the two definitions of the friction coefficient in Eq. (2.2). (a) Three examples of friction-load curves consisting of a typical linear curve (blue), a linear curve with a shift representing an adhesive contact (red), and a non-linear curve (purple). The corresponding friction coefficients μ_1 and μ_2 are shown for the first definition Eq. (2.2a) in (b) and the second definition Eq. (2.2a) in (c).

Amontons’ law represents the frictional behavior relatively accurately for many surfaces in contact, involving both dry and lubricated, ductile and brittle and rough and smooth surfaces (as long as they are not adhesive) under a variety of conditions [27]. But it has its limitations. For instance, at low velocities, Amontons’ model breaks down due to thermal effects, and for high velocities due to inertial effects [1, pp. 5–6]. Additionally, static friction depends on the so-called contact history, with increasing static friction as the logarithm of time in stationary contact [28].

In cases where Amontons’ law breaks down, we might still use the conceptual definition of the friction coefficient as defined by Eq. (2.2b). Especially, in the context of achieving negative friction coefficients (for certain load ranges), we would refer to this definition, since Eq. (2.2a) would imply a truly unphysical situation of the friction force acting in the same direction as the sliding motion. This would accelerate the object indefinitely².

Due to the empirical foundation of Amontons’ law, it does not provide any physical insight into the underlying mechanisms of friction. However, as we will later discuss in more detail, we can understand the overall phenomena of friction through statistical mechanics by the concept of *equipartition of energy* [24]. A

²You would most likely have a good shot at the Nobel Prize with that paper.

system in equilibrium has its kinetic energy uniformly distributed among all its degrees of freedom. When a macroscale object is sliding in a given direction it is clearly not in equilibrium since one of its degrees of freedom carries considerably more kinetic energy. Thus, the system will tend to transfer kinetic energy to the remaining degrees of freedom in the form of heat dissipation to the surroundings. This will make the object slow down if not continuously driven forward by an external energy source. Hence, we can understand the overall concept of friction simply as the tendency towards energy equipartitioning among many interacting degrees of freedom [24]. From this point of view, it is clear that friction is an inevitable part of contact physics, but even though friction cannot be removed altogether, we are still capable of manipulating it in useful ways.

The attentive reader might point out that we have already moved the discussion into the microscopic regime as *statistical mechanics* generally aim to explain macroscale behavior by microscopic interactions. This highlights the necessity to consider smaller scales in order to achieve a more fundamental understanding of friction.

We note that more advanced models for macroscale friction exist. For instance, the earthquake-like (EQ) model, also known as the *spring-and-block* model or the *multi-contact* model [24], developed by Burridge and Knopoff [29]. This has been used in many studies of earthquake friction [30] and similar schemes have since been used to model the failure of fiber bundles and faults [31, 32]. Also, *rate and state* models have been used for such macroscale modeling [33]. However, these extensions are beyond the scope of this thesis as we will mainly focus on the nanoscale description.

2.3 Microscopic scale

Going from a macro- to a microscale perspective, at a length scale on the order of 10^{-6} m, it was realised that most surfaces are in fact rough [34]. The contact between two surfaces consists of numerous smaller contact points, so-called *asperities*, which form junctions due to contact pressure and adhesion as visualized in Fig. 2.2 [3]. In the macroscale perspective of Amonton's law, we refer to time- and space-averaged values, i.e. the apparent contact area and the average sliding speed [27]. However, microscopically we find the real contact area to be much smaller than the apparent area [3], and the shearing motion of local microjunctions to happen at large fluctuations rather than as one synchronized movement throughout the surface.

It is generally accepted that friction is caused by two mechanisms: Mechanical friction and chemical friction [3]. Mechanical friction is the “plowing” of the surface by hard particles or said asperities with an energy loss attributed to deformations of the asperities. While plastic deformations, corresponding to wear, gives rise to an obvious attribution for the energy loss, elastic deformations are also sufficient in explaining energy loss due to phonon excitations. The assumption of plastic deformations has been criticized as this is theorized only to be present at the beginning of a surface contact while it is negligible for prolonged or repeated contacts [35]. That is, when machine parts slide against each other for millions of cycles, the plastic deformation would only take place at the beginning for which the system then reaches a steady state with only elastic deformations. The chemical friction arises from adhesion between microscopic contacting surfaces, with an energy loss attributed to the breaking and forming of chemical bonds between the interacting surfaces.

2.3.1 Asperity theories

Asperity theories have their foundations in the adhesion model proposed by Bowden and Tabor [36] which is based on the fundamental reasoning that friction is governed by the adhesion between two surfaces [37]. Adhesion is proportional to the real contact area defined by asperity junctions, and interfacial shear strength τ between such contacting junctions. For an asperity contact area A_{asp} we get a true contact area $\sum A_{\text{asp}}$ leading to

$$F_{\text{fric}} = \tau \sum A_{\text{asp}}.$$

Note that this is still compatible with Amontons' law in Eq. (2.1) by having a linear relationship between the real contact area and the applied load. By increasing the normal load it is hypothesized that the real contact area will increase as the asperity tips are deformed (plastically or elastically) into broader contact points as visualized qualitatively in Fig. 2.2.



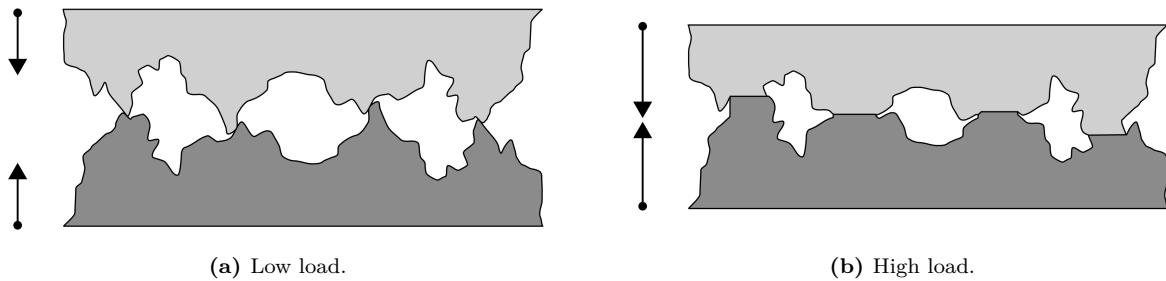


Figure 2.2: Qualitatively illustration of the microscopic asperity deformation under increasing load from frame (a) to (b). While this figure seemingly portrays plastic deformation the concept of increased contact area with increased load applies to elastic deformation as well. Reproduced from [38].

Many studies have focused on single asperity contacts to reveal the relationship between the contact area and load [39–41]. By assuming perfectly smooth asperities, with radii of curvature from micrometers all the way down to nanometers, continuum mechanics can be used to predict the deformation of asperities as load is applied. A model for non-adhesive contact between homogenous, isotropic, linear elastic spheres was first developed by Hertz [42], which predicted $A_{\text{asp}} \propto F_N^{2/3}$. Later adhesion effects were included in a number of subsequent models, including Maugis-Dugdale theory [43], which also predicts a sublinear relationship between A_{asp} and F_N . Thus, the common feature of all single-asperity theories is that A_{asp} is a sublinear function of F_N , leading to a similar sublinear relationship for $F_{\text{fric}}(F_N)$. This fails to align with the macroscale observations modeled by Amontons' law (Eq. (2.1)).

Concurrently with single-asperity studies, roughness contact theories are being developed [44–47] to bridge the gap between single asperities and macroscopic contacts [34]. A variety of multi-asperity theories has attempted to combine single asperity mechanics by statistical modeling of the asperity height and spatial distributions [35]. This has led to partial success in the establishment of a linear relationship between A_{asp} and F_N . Unfortunately, these results are restricted in terms of the magnitude of the load and contact area, where multi-asperity contact models based on the original ideas of Greenwood and Williamson [46] only predicts linearity at vanishing loads, or Persson [45] which predicts linearity for more reasonable loads up to 10–15% of the macroscale contact area. However, as the load is further increased all multi-asperity models predict the contact area to fall into the sublinear dependency of normal force as seen for single asperity theories as well [35].

2.4 Nanoscale — Atomic scale

Going from a micro- to a nanoscale, on the order of 10^{-9} m, it has been predicted that continuum mechanics will start to break down [48] due to the discreteness of individual atoms. In a numerical MD study by Mo et al. [34], considering asperity radii of 5–30 nm, it has been shown that the asperity area A_{asp} , defined by the circumference of the contact zone, is sublinear with F_N . This is accommodated by the observation that not all atoms within the circumference make chemical contact with the substrate. By modeling the real contact area $A_{\text{real}} = N A_{\text{atom}}$, where N is the number of atoms within the range of chemical interaction and A_{atom} the associated surface area for a contacting atom, they found a consistent linear relationship between friction and the real contact area. Without adhesive forces, this leads to a similar linear relationship $F_{\text{fric}} \propto F_N$, while adding van der Waals adhesion to the simulation gave a sublinear relationship matching microscale single asperity theory, even though the $F_{\text{fric}} \propto A_{\text{real}}$ was maintained. This result emphasizes that the predictions of continuum mechanisms might still apply at the nanoscale and that the contact area can be expected to play an important role in nanoscale asperity contacts. It is simply the definition of the contact area that changes when transitioning from the microscale to the nanoscale.

While the study by Mo et al. [34] considers a single asperity on a nanoscale, some models take this even further to what we will denote as the atomic scale. This final leap is motivated by the fact that our system of interest, an atomically flat graphene sheet imposed on a flat silicon substrate, lacks the presence of nanoscale asperities in its initial uncut undeformed state. In the lack of noteworthy structural asperities, friction can instead be modeled as a consequence of the “rough” potential laid out by the atomic landscape. A series of so-called *reduced-order* models build on a simplified system of atomic-scale contacts based on three essential parts: 1) A periodic potential modeling the substrate as a rigid crystalline surface. 2) An interacting particle, or collection of particles, placed in the potential. 3) A moving body, moving at a steady speed, connected to the particles through a harmonic spring. In figure Fig. 2.3 three of the most common 1D models are displayed

which we will address in the following sections. The time-honored Prandtl-Tomlinson (PT) model describes a point-like tip sliding over a space-periodic fixed crystalline surface with a harmonic coupling to the moving body. This is analog to that of an experimental cantilever used for Atomic Force Microscopy which we will introduce in more detail in Sec. 2.4.5.1. Further extensions were added in the Frenkel-Kontorova (FK) model by substituting the tip with a chain of harmonically coupled particles dragged from the end, and finally combined in the Frenkel-Kontorova-Tomlinson (FKT) with the addition of a more rigorous harmonic coupling between the moving body and each of the atoms in the chain. While these models cannot provide the same level of detail as atomistic simulations, such as MD, they enable investigation of atomic friction under most conditions, some of which are inaccessible to MD [49]. This makes these models an appropriate tool for investigating individual parameters and mechanisms governing friction.

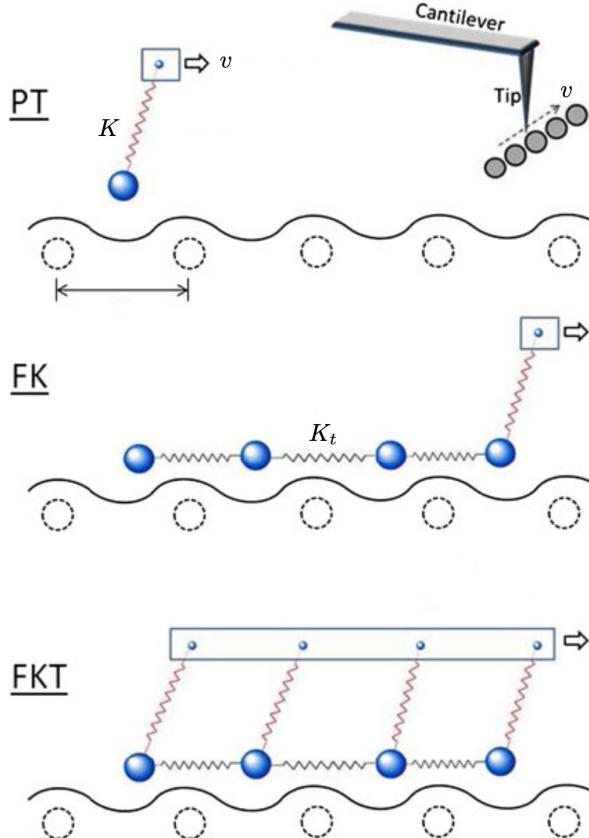


Figure 2.3: Illustration of the key features of the Prandtl-Tomlinson (PT), Frenkel-Kontorova (FK) and Frenkel-Kontorova-Tomlinson (FKT) reduced-order models respectively. Reproduced from [49] with modifications of the displayed notation.

2.4.1 Prandtl–Tomlinson

The Prandtl–Tomlinson model (PT) considers a 1D simplification of the frictional system as a single ball-tip sliding along the rigid substrate as shown in Fig. 2.3. The tip is coupled harmonically to a moving support, moving at a constant speed, which drives the tip forward. The interaction between the tip and the substrate is modeled by a sinusoidal corrugation potential mimicking the periodicity found in a crystalline substrate. We will consider the Prandtl–Tomlinson model with added thermal activation as proposed by Gnecco et al. [50]. For the theoretical foundation of this section, we generally refer to [49]. The potential energy for the tip at position x at time t is given as

$$V(x, t) = \frac{1}{2}K(vt - x)^2 - \frac{1}{2}U_0 \cos\left(\frac{2\pi x}{a}\right). \quad (2.3)$$

The first term describes the harmonic coupling with spring constant K , between the tip at position x and the moving body at position vt , given by its constant speed v . The second term describes the corrugation potential with amplitude U_0 and period a representing the lattice spacing of the substrate. The dynamics of



the tip can be described by the Langevin equations

$$m\ddot{x} + m\mu\dot{x} = -\frac{\partial V(x, t)}{\partial x} + R(t), \quad (2.4)$$

where m is the mass of the tip, μ the viscous friction and $R(t)$ the thermal activation term. The equation is solved for tip position x and the friction force is retrieved as the force acting on the moving body

$$F_{\text{fric}} = K(vt - x).$$

The governing equation Eq. (2.4) belongs to a family of stochastic differential equations composed of both deterministic dynamics and stochastic processes. In this case, the deterministic term is the viscous friction, $m\mu\dot{x}$, resisting the movement of the tip. The stochastic term is a random force field modeling thermal noise according to the Fluctuation-dissipation relation. Thus, there is no single path but rather multiple paths the tip can take. While the Langevin equation is one of the most common ways to handle thermal activation other methods exist to solve this problem such as Monte Carlo sampling methods. We omit the numerical scheme for the solving of the Langevin equations here and refer instead to a more in-depth discussion of the Langevin equation regarding the use in MD simulations in Sec. 3.3.1.

2.4.1.1 Thermal activation

The solving of the Langevin equation, as opposed to Newton's equation of motion, introduces thermal effects to the system. Generally, when the energy barrier comes close to $k_B T$ (0.026 eV at room temperature) thermal effects can not be neglected [49]. In the case of a single asperity contact the energy barrier is on the order 1 eV which makes thermal activation significant. Due to the moving body traveling at a constant speed, the potential energy will increase steadily. Without any temperature, $T = 0$, the slip will only occur when the energy barrier between the current potential well i and the adjacent j is zero $\Delta V_{i \rightarrow j} = 0$. However, in the presence of temperature, we get thermal activation, meaning that the tip can slip to the next potential well sooner at $\Delta V_{i \rightarrow j} > 0$. Provided that the sliding speed is slow enough the transition rate κ for a slip from the current to the next well is given by

$$\kappa = f_0 e^{-\Delta V/k_B T}, \quad (2.5)$$

with ΔV being the energy barrier and f_0 the attempt rate. The attempt rate following Kramer's rate theory [51] is related to the mass and damping of the system and can be thought of as the frequency at which the tip "attempts" to overcome the barrier. Notice that Eq. (2.5) resembles a microstate probability in the canonical ensemble with f_0 in place of the inverse partition function Z^{-1} which provides an additional interpretation of f_0 . The probability p_i that the tip occupies the current well i relative to the adjacent well j , as illustrated in Fig. 2.4, is governed by

$$\frac{dp_i}{dt} = -\kappa_{i \rightarrow j} p_i + \kappa_{j \rightarrow i} p_j. \quad (2.6)$$

This probability is related to temperature, speed and mass [49].

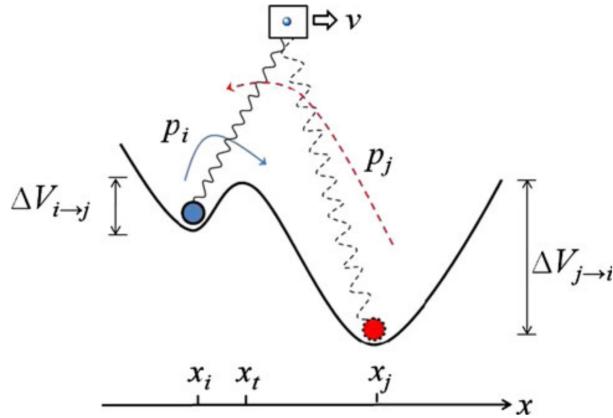


Figure 2.4: An illustration of slip between two adjacent energy minima. p_i is the probability of the tip residing in the current potential well, i , where the energy barrier is $\Delta V_{i \rightarrow j}$. p_j is the probability of the tip residing at the next minima, j , where $\Delta V_{j \rightarrow i}$ is the corresponding energy barrier. Figure and caption reproduced from [49].

Generally, there exist two temperature regimes in the Prandtl–Tomlinson model: The *thermal activation* regime at low temperatures and the *thermal drift* at high temperatures as shown in Fig. 2.5. At lower temperatures, the system is subject to standard thermal activation with a much lower energy barrier for slipping forward than backward $\Delta V_{j \rightarrow i} \gg \Delta V_{i \rightarrow j}$. This results in a higher transition rate for forward slips, $\kappa_{j \rightarrow i} \ll \kappa_{i \rightarrow j}$, which effectively inhibits any backward slips and simplifies Eq. (2.6) to

$$\frac{dp_i}{dt} = -\kappa_{i \rightarrow j} p_i.$$

This leads to the relationship between friction, temperature and speed following Sang et al.'s prediction [52]

$$F_{\text{fric}} = F_c - \left| \beta k_B T \ln \left(\frac{v_c}{v} \right) \right|^{2/3}, \quad v_c = \frac{2f_0 \beta k_B T}{3C_{\text{eff}} \sqrt{F_c}}, \quad (2.7)$$

where F_c is the maximum friction at $T = 0$, v_c a critical velocity, f_0 is the attempt rate, c_{eff} the effective stiffness, and β a parameter determined by the shape of the corrugation well. Eq. (2.7) characterizes the decrease in friction with temperature in the thermal activation regime, shown in Fig. 2.5a at low temperature. This corresponds with the assumption of only forward slips, as seen in the force trace in Fig. 2.5a. When the temperature is high enough for the system to be consistently close to thermal equilibrium, it enters the regime of thermal drift [53]. This regime transition can be understood through a comparison between two time scales: The time it takes for the moving body to travel one lattice spacing $t_v = a/v$ and the average time for a slip to occur due to thermal activation $\tau = 1/\kappa = f_0^{-1} \exp(\Delta V/k_B T)$. If $t_v \gg \tau$ the system falls into the thermal drift regime, where slips happen both in the forward and backward direction as shown in the force trace in Fig. 2.5b. For the thermal drift regime, the friction follows the prediction by Krylov et al. [53–55]

$$F_{\text{fric}} \propto \frac{v}{T} e^{1/T}. \quad (2.8)$$

Notice that the friction dependence on sliding speed also changes from Eq. (2.7) to Eq. (2.8) as it transitions from the thermal activation to the thermal drift regime.

2.4.1.2 Sliding speed

In the thermal activation regime (low temperature) and at low sliding speeds, the friction relation follows Eq. (2.7) which means that friction increases logarithmically with speed. For higher speeds, above the critical velocity $v > v_c$, if only thermal effects are considered, Eq. (2.7) predicts that friction will eventually saturate and come to a plateau at $F_{\text{fric}} = F_c$. This is illustrated in Fig. 2.6 with this prediction being represented by the dotted line. However, as given away by the figure, for higher speeds the model will enter an *athermal* regime where the thermal effects are negligible compared to other contributions [56]. In the athermal regime, the damping term $m\mu\dot{x}$ will dominate yielding $F_{\text{fric}} \propto v$. The athermal regime is often observed in reduced-models if the system is overdamped or at high speeds. This concept is related to MD simulations as well where the accessible speeds often fall into the athermal regime [57]. It is unclear how this affects real physical systems for which there exist more dissipation channels than just a single viscous term [58]. For the thermal drift regime, at higher temperatures, friction increase linearly with sliding speed $F_{\text{fric}} \propto v$ as given by Eq. (2.8).

2.4.1.3 Tip mass

The mass of the tip affects the dynamics due to a change of inertia, which changes the attempt rate f_0 . Smaller inertia leads to a larger attempt rate and vice versa. Effectively, this will affect the transition point for the temperature and speed regimes described previously. A smaller inertia, yielding a larger attempt rate, will cause an earlier transition, i.e. at a lower temperature, to the thermal drift regime. Additionally, this will also result in a later transition to the athermal regime, i.e. at a higher speed.

2.4.1.4 Friction regimes: Smooth sliding, single slip, and multiple slip

Stick-slip motion is a crucial instability mechanism associated with high energy dissipation and high friction. Thus, controlling the transition between smooth sliding and stick-slip is considered key to controlling friction. We can divide the frictional stick-slip behavior into three regimes: 1) Smooth sliding, where the tip slides smoothly on the substrate. 2) Single slip, where the tip stick at one potential well before jumping one lattice spacing to the next. 3) Multiple slip, where the tip jumps more than one lattice spacing in a slip event. The underlying mechanisms behind these regimes can be understood through static and dynamic contributions.

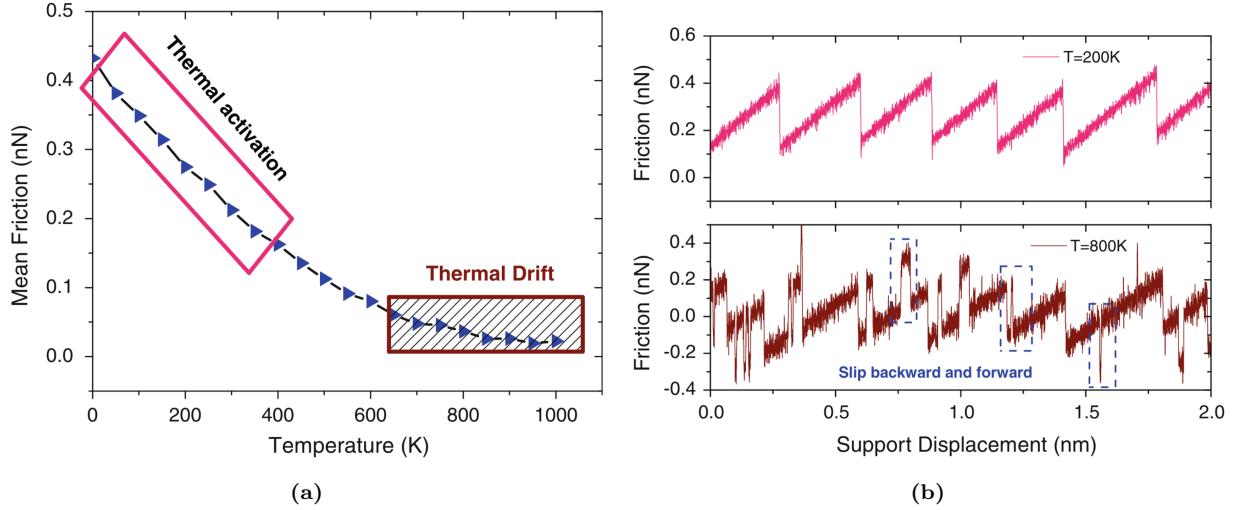


Figure 2.5: Illustration of the temperature difference between the thermal activation regime and the thermal drift regime. (a) The mean friction as a function of temperature showcasing the regime transition. The figure corresponds to the numerical results of Dong et al. [49] of a Prandtl–Tomlinson model with model parameters: $m = 10^{-12}$ kg, $U_0 = 0.6\text{eV}$, $v = 4 \times 10^3$ nm/s, $\mu = 2\sqrt{\text{K}/\text{m}}$, $a = 0.288$ nm. (b) The force traces of a system in the thermal activation regime (top) and thermal drift regime (bottom) with several characteristic forward and backward slips highlighted by dashed lines. The forward slips are identified as a sudden decrease in friction while the backward slips are identified as a sudden increase in friction. Reproduced from [49].

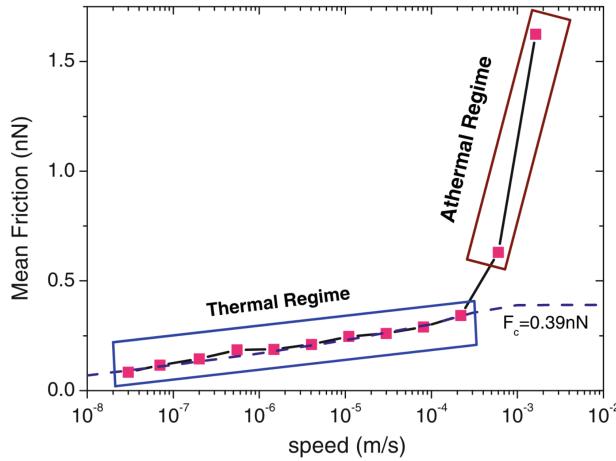


Figure 2.6: The friction dependence on sliding speed for the simulated Prandtl–Tomlinson model by Dong et al. [49] in the thermal activation temperature regime, revealing two different sliding speed regimes. In the thermal regime, friction increases logarithmically with sliding speed following Eq. (2.7), and in the athermal regime, friction is governed by damping leading to a proportional relationship to sliding speed $F_{\text{fric}} \propto v$. The friction plateau ($F_c = 0.39$ nN) predicted by thermal activation Eq. (2.7) is shown as a dotted line. Other models parameters: $m = 10^{-12}$ kg, $U_0 = 0.6\text{eV}$, $T = 300$ K, $\mu = 2\sqrt{\text{K}/\text{m}}$, $a = 0.288$ nm. Reproduced from [49].

To understand the static mechanism we consider a quasistatic process for which temperature, speed and damping can be neglected. For a quasistatic process, we require $\partial V/\partial x = 0$. This simplifies Eq. (2.3) to

$$\frac{\pi U_0}{a} \sin\left(\frac{2\pi x}{a}\right) = K(vt - x). \quad (2.9)$$

The friction regime is determined by the number of solutions x to Eq. (2.9). Only one solution corresponds to smooth sliding, two solutions to a single slip and so on. It turns out that the regimes can be defined by the parameter $\eta = 2\pi^2 U_0 / a^2 K$ [59, 60] yielding transitions at $\eta = 1, 4.6, 7.79, 10.95, \dots$, such that $\eta \leq 1$ corresponds to smooth sliding, $1 < \eta \leq 4.6$ to a single slip and so on. These static derivations lay out the fundamental probabilities for being in one of the stick-slip regimes. Notice that increasing the spring

constant K (stiff spring) will decrease the probability of stick-slip behavior. Similarly, η can be altered by a change in the potential corrugation U_0 through an increased load [61].

Considering the dynamics on top, one finds that damping, speed and temperature will affect this probability. High damping, equivalent to a high transfer of kinetic energy to heat, will result in less energy available for the slip events. This will make multiple slip events less likely. By a similar argument, we find that increasing the speed will contribute to more kinetic energy which will increase the likelihood of multiple slip events. Finally, the temperature will contribute to earlier slips, due to thermal activation, such that less potential energy can be accumulated and it will result in fewer multiple slip events.

2.4.2 Frenkel-Kontorova

The Frenkel-Kontorova (FK) model [62] extends the Prandtl–Tomlinson model by considering a chain of atoms in contrast to just a single particle (tip). This extension is useful for understanding the importance of the alignment between the atoms and the substrate, the so-called *commensurability*. Our review of the Frenkel-Kontorova is based on [24, 61].

The standard Frenkel-Kontorova model consists of a 1D chain of N classical particles of equal mass, representing atoms, interacting via harmonic forces and moving in a sinusoidal potential as sketched in Fig. 2.7 [24]. The Hamiltonian is

$$H = \sum_{i=1}^N \left[\frac{p_i^2}{2m} + \frac{1}{2}K(x_{i+1} - x_i - a_c)^2 + \frac{1}{2}U_0 \cos\left(\frac{2\pi x_i}{a_b}\right) \right], \quad (2.10)$$

where the atoms are labelled sequently $i = 1, \dots, N$. The first term $p_i^2/2m$ represents the kinetic energy with momentum p_i and mass m . Often the effects of inertia are neglected, referred to as the static Frenkel-Kontorova model, while the inclusion in Eq. (2.10) is known as the dynamic Frenkel-Kontorova model [63]. The next term describes the harmonic interaction with elastic constant K , nearest neighbor distance $\Delta x = x_{i+1} - x_i$ and corresponding nearest neighbor equilibrium distance a_c . The final term represents the periodic corrugation potential, with amplitude U_0 and period a_b . By comparison to the potential used in the Prandtl–Tomlinson model Eq. (2.3), we find the difference to be the introduction of a harmonic coupling between particles in the chain. Notice also, that we have not yet specified the motion of the connected moving body. Different boundary choices can be made where both free ends and periodic conditions give similar results. The choice of fixed ends however makes the chain incapable of sliding.

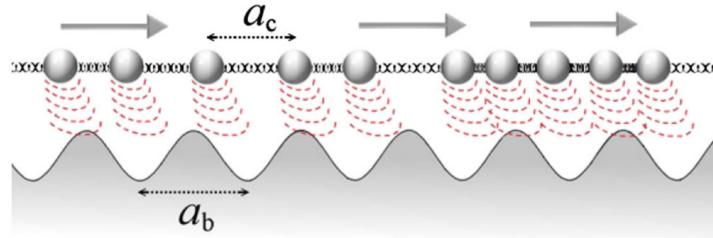


Figure 2.7: An illustration of the Frenkel-Kontorova model with two competing lengths: The interparticle distance a_c and the substrate periodicity a_b . Figure reproduced from [61] with permission from the American Physical Society.

To probe static friction one can apply an external adiabatically increasing force, i.e. without loss or gain of heat, until sliding occurs. This corresponds to the static Frenkel-Kontorova model, and it turns out that the sliding properties are entirely governed by its topological excitations referred to as so-called *kinks* and *antikinks*.

2.4.2.1 Commensurability

We can subdivide the frictional behavior in terms of commensurability, that is, how well the spacing of the atoms matches the periodicity of the substrate potential. We describe this by the length ratio $\theta = a_b/a_c = N/M$ where M denotes the number of minima in the potential within the length of the chain. A rational number for θ means that we can achieve a perfect alignment between the atoms in the chain and the potential minima, without stretching the chain, corresponding to a *commensurate* case. If θ is irrational the chain and substrate cannot fully align without some stretching of the chain, and we denote this as being *incommensurate*.



We begin with the simplest commensurate case of $\theta = 1$ where the spacing of the atoms matches perfectly with the substrate potential periodicity, i.e. $a_c = a_b$, $N = M$. The ground state (GS) is the configuration where each atom is aligned with one of the substrate minima. By adding an extra atom to the chain we would effectively shift some of the atoms out of this ideal state, giving rise to a kink excitation. This leads to the case where two atoms will have to “share” the same potential corrugation as sketched in Fig. 2.8. On the other hand, removing an atom from the chain results in an antikink excitation where one potential corrugation will be left “atomless”. In order to reach a local minimum the kink (antikink) will expand in space over a finite length such that the chain undertakes a local compression (expansion). Notice that for low ratios of θ , there are fewer atoms than minima, and the chain will not be able to fill each corrugation well. In this case, the kink excitations can instead be thought of as whether the atoms are forced to be in a different potential well than otherwise dictated by the spring forces in-between.



Figure 2.8: Qualitative example of an incommensurable case where the atoms sit slightly closer together than otherwise dictated by the substrate periodicity. This results in a single kink which is here seen as the presence of two atoms within the same potential corrugation well. Reproduced from [64].

When applying a tangential force to the chain it is much easier for an excitation to move along the chain than it is for the non-excited atoms since the activation energy for a kink/antikink displacement is systematically smaller (often much smaller) than the potential barrier U_0 . Thus, the motion of kinks (antikinks), i.e. the displacement of extra atoms (atom vacancies), is representing the fundamental mechanism for mass transport. These displacements are responsible for the mobility, diffusivity and conductivity within this model. In the zero temperature commensurable case with an adiabatical increase in force, all atoms would be put into an accelerating motion as soon as the potential barrier energy is present. However, similar to our discussion on the Prandtl-Tomlinson model, thermal activations will excite the system at an earlier stage resulting in kink-antikink pairs traveling down the chain. For a non-periodic chain of finite length, these often occur at the end of the chain running in opposite directions. This cascade of kink-antikink excitations is shown in Fig. 2.9. Notice, that for the 2D case, where an island (flake) is deposited on a surface, we generally also expect the sliding to be initiated by kink-antikink pairs at the boundaries.

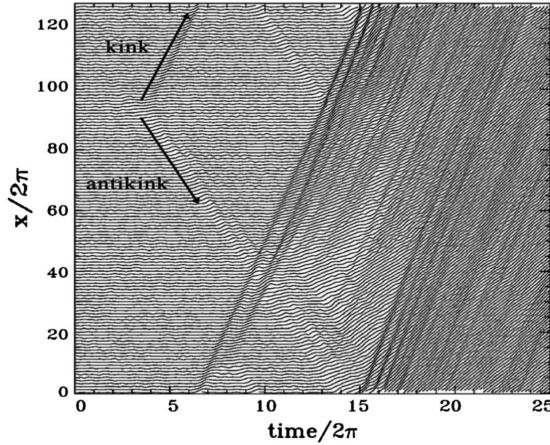


Figure 2.9: Atomic trajectories vs. time at the depinning transition for a periodic nonzero temperature Frenkel-Kontorova chain with $\theta = 1$. The onset of motion is marked by the creation of one kink-antikink pair. The kink and antikink move in opposite directions. They collide after passing through the boundaries, and soon a second kink-antikink pair is created in the tail of the primary kink. This process repeats with exponential avalanche-like growth of the kink-antikink concentration, leading to the total sliding state. Adapted from [65], figure from [61] with permission from the American Physical Society.

For the case of incommensurability, i.e. $\theta = a_b/a_c$ being irrational, the GS is characterized by a sort of “staircase” deformation. That is, the chain will exhibit regular periods of regions with approximate commensurability separated by regularly spaced kinks or antikinks.

The incommensurable Frenkel-Kontorova model contains a critical elastic constant K_c , such that for $K > K_c$ the static friction F_s drops to zero, making the chain able to initiate a slide at no energy cost,

while the low-velocity kinetic friction is dramatically reduced. This can be explained by the fact that the displacement occurring in the incommensurable case will yield just as many atoms climbing up a corrugation as atoms climbing down. For a big (infinite) chain this will exactly balance the forces making it non-resistant to sliding. Generally, incommensurability guarantees that the total energy (at $T = 0$) is independent of the relative position to the potential. However, when sliding freely, a single atom will eventually occupy a maximum of the potential, and thus when increasing the potential magnitude U_0 or softening the chain stiffness, lowering K , the possibility to occupy such a maximum disappears. This marks the so-called *Aubry transition*, at the critical elastic constant $K = K_c(U_0, \theta)$, where the chain goes from a free sliding to a *pinned* state with nonzero static friction. K_c is a discontinuous function of the ratio θ , due to the reliance on irrational numbers for incommensurability. The minimal value $K_c \simeq 1.0291926$ in units $[2U_0(\pi/a_b)^2]$ is achieved for the golden-mean ratio $\theta = (1 + \sqrt{5}/2)$. The Aubry transition can be investigated as a first-order phase transition for which power laws can be defined for the order parameter, but this is beyond the scope of this thesis.

The phenomenon of non-pinned configurations is named *superlubricity* in tribological context. Despite the misleading name, this refers to the case where the static friction is zero while the kinetic friction is nonzero, but reduced. For the case of a 2D sheet, it is possible to alter the commensurability, not only by changing the lattice spacing through material choice but also by changing the orientation of the sheet relative to the substrate. Dienwiebel et al. [66] have observed experimentally that the kinetic friction, for a graphene flake sliding over a graphite surface (multiple layers of graphene), exhibits extremely low friction at certain orientations as shown in Fig. 2.10. As the orientation is changed they observed two spikes of considerable friction while the remaining valleys correspond to effectively zero friction in consideration of the measurement uncertainty. This phenomenon relates to the transition between frictional slip regimes, as introduced through the Prandtl–Tomlinson model, since the change in orientation affects the effective substrate potential. Merely from the static consideration, we found that lowering the potential amplitude U_0 will decrease the parameter $\eta = 2\pi^2 U_0/a^2 K$ shifting away from the regime of multiple slip towards smooth sliding associated with low friction. Such transitions will also be affected by the shape of the potential and corresponding 2D effects of the sliding path [49].

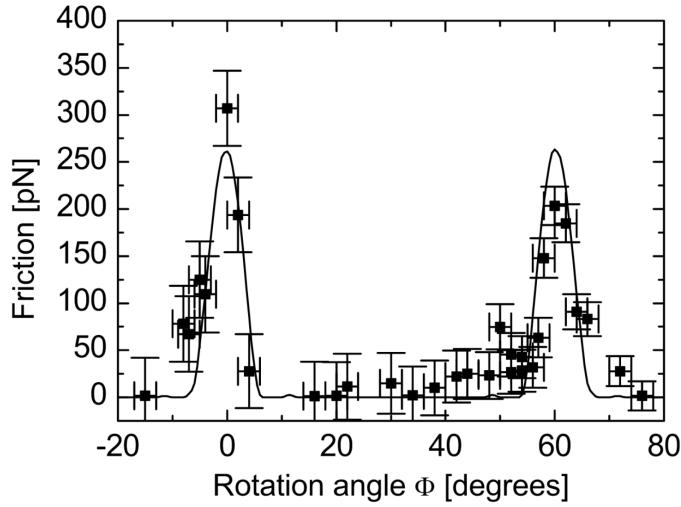


Figure 2.10: Average friction force versus rotation angle Φ of the graphite sample around an axis normal to the sample surface. Two narrow peaks of high friction are observed at 0° and 61° , respectively. Between these peaks, a wide angular range with ultra-low friction, close to the detection limit of the instrument, is found. The first peak has a maximum friction force of 306 ± 40 pN, and the second peak has a maximum of 203 ± 20 pN. The curve through the data points shows results from a Tomlinson model for a symmetric 96-atom graphite flake sliding over the graphite surface (for details about the calculation see [67]). Figure and caption adapted from [66], reproduced from [61] with permission from the American Physical Society.

2.4.2.2 Velocity resonance

While many of the same arguments used for the Prandtl–Tomlinson model regarding velocity dependence for friction can be made for the Frenkel-Kontorova model as well, the addition of multiple atoms introduces the possibility of resonance. In the Frenkel-Kontorova model, the kinetic friction is primarily attributed to



resonance between the sliding-induced vibrations and phonon modes in the chain [63]. The specific dynamics are found to be highly model and dimension specific, and even for the 1D case, this is rather complex. However, we make a simplified analysis of the 1D rigid chain in order to showcase the reasoning behind the phenomenon.

When all atoms are sliding rigidly with center of mass (CM) velocity v_{CM} the atoms will pass the potential maxima with the so-called *washboard frequency* $\Omega = 2\pi v_{\text{CM}}/a_b$. For a weak coupling between the chain and the potential we can use the zero potential case as an approximation for which the known dispersion relation for the 1D harmonic chain is given [68, p. 92]

$$\omega_k = \sqrt{\frac{4K}{m}} \left| \sin\left(\frac{k}{2}\right) \right|,$$

where ω_k is the phonon frequency and $k = 2\pi i/N$ the wavenumber with $i \in [N/2, N/2)$. Resonance will occur when the washboard frequency Ω is close to the frequency of the phonon modes ω_q in the chain with wavenumber $q = 2\pi a_c/a_b = 2\pi\theta^{-1}$ or its harmonics nq for $n = 1, 2, 3, \dots$ [69]. Thus, we can approximate the resonance CM speed as

$$\begin{aligned} n\Omega &\sim \omega_{nq} \\ n\frac{2\pi v_{\text{CM}}}{a_b} &\sim \sqrt{\frac{4K}{m}} \left| \sin\left(\frac{2n\pi\theta^{-1}}{2}\right) \right| \\ v_{\text{CM}} &\sim \frac{\sin(n\pi\theta^{-1})}{n\pi} \sqrt{\frac{Ka_b^2}{m}}. \end{aligned}$$

When the chain slides with a velocity around resonance speed, the washboard frequency can excite acoustic phonons which will dissipate to other phonon modes as well. At zero temperature, the energy will transform back and forth between internal degrees of freedom and CM movement of the chain. Without any dissipation mechanism, this is theorized to speed up the translational decay [63]. However, as soon as we add a dissipation channel through the substrate, energy will dissipate from the chain to the substrate's degrees of freedom. This suggests that certain sliding speeds will exhibit relatively high kinetic friction while others will be subject to relatively low kinetic friction. Simulations of concentric nanotubes in relative motion (telescopic sliding) support this idea as it has revealed the occurrence of certain velocities at which the friction is enhanced, corresponding to the washboard frequency of the system [70, 71]. The friction response was observed to be highly non-linear as the resonance velocities were approached.

The analysis of the phonon dynamics is highly simplified here, and a numerical study of the 2D Frenkel-Kontorova model by Norell et al. [63] showed that the behavior was highly dependent on model parameter choices, but that the friction generally increased with velocity and temperature. This temperature dependence differs qualitatively from that of the Prandtl–Tomlinson model.

2.4.3 Frenkel-Kontorova-Tomlinson

A final extension of the reduced-order models worth mentioning here is the Frenkel-Kontorova-Tomlinson (FKT) model [72], which introduces a harmonic coupling between the moving body and each of the atoms in the sliding chain, effectively combining the Prandtl–Tomlinson and Frenkel-Kontorova models (see Fig. 2.3). This introduces more degrees of freedom to the model which is based on the intention of achieving a more realistic modeling of the connection between the moving body and the chain. Dong et al. [49] carried out a numerical analysis using the 1D Frenkel-Kontorova-Tomlinson model to investigate the effect of chain length. They observed that the friction generally increased linearly with the number of atoms in the chain on a long range, but certain lattice mismatches resulted in local non-linear relationships as shown in Fig. 2.11a. Similarly, by extending the Frenkel-Kontorova-Tomlinson model to 2D they were able to achieve a similar sensitivity to commensurability as observed experimentally by [66] (see Fig. 2.10). This numerical result is shown in Fig. 2.11b. Besides a demonstration of the commensurability effect in 2D they also observed increasing friction with an increasing flake size. Combined, the 1D and 2D results support the idea of increasing friction with contact size although it might showcase non-linear behavior depending on commensurability.

2.4.4 Shortcomings of the reduced-models

It should be noted that the reduced-models presented in the previous sections provide a simplified description of the friction behavior. One major limitation is that all models assume a rigid substrate with a constant sinusoidal potential shape. In reality, the potential shape might be more complex and also dynamically

changing as the substrate reacts to the sliding motion, resulting in a more intricate system. Moreover, the energy dissipation is simplified through a viscous term $-m\mu\dot{x}$ in the Langevin equation Eq. (2.4), which neglects the complexity associated with electron and phonon dissipation. For example, considering phonon dissipation, there exist many vibration modes ($3N$), and thus many dissipation channels for the tip [49]. Lastly, it should be mentioned that the moving body is assumed to move rigidly with constant speed, whereas, in reality, it may exhibit a more complicated dynamic behavior.

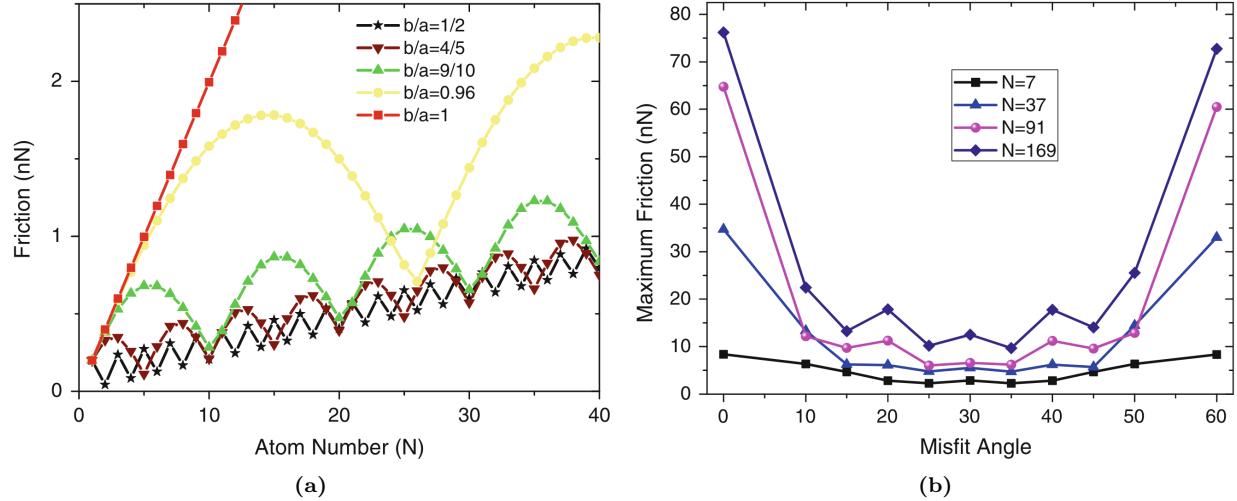


Figure 2.11: Friction in the Frenkel-Kontorova-Tomlinson model for varying size and commensurability corresponding to the numerical result by Dong et al. [49]. The spring constant is K_t for the interatomic coupling and K for the coupling to the moving body. (a) The 1D case with an increasing number of atoms in the chain and different mismatch length ratios $\theta = a_b/a_c$. The figure notation corresponds to $a_b = a$ and $a_c = b$ yielding $b/a = \theta^{-1}$. The model parameters are $K = 5 \text{ N/m}$ and $K_t = 50 \text{ N/m}$. (b) The 2D case with varying angles (misfit angle) between the flake and the substrate. The model parameters are $K = 10 \text{ N/m}$ and $K_t = 50 \text{ N/m}$. Reproduced from [49].

2.4.5 Experimental procedures

Experimentally, the study of nanoscale friction is challenging due to the low forces on the scale of nano-newtons along with the difficulties of mapping the nanoscale topography of the sample. In contrast to numerical simulations, which provide full transparency regarding atomic-scale structures, sampling of forces, velocities and temperature, the experimental results are limited by the state-of-the-art experimental methods. To facilitate the comparison of numerical and experimental results, we will address a few of the most relevant experimental methods.

2.4.5.1 Scanning Probe Microscopy

Scanning probe microscopy (SPM) includes a variety of experimental methods which are used to examine surfaces with atomic resolution [73, pp. 6–27]. This was originally developed for surface topography imaging, but today it plays a crucial role in nanoscale science as it is used for probe-sampling regarding tribological, electronic, magnetic, biological and chemical character. The family of methods involving the measurement of forces is generally referred to as *scanning force microscopy* (SFM) or for friction purposes *friction force microscopy* (FFM).

One such method arose from the *atomic force microscope* AFM, which consists of a sharp micro-fabricated tip attached to a cantilever force sensor, usually with a sensitivity below 1 nN all the way down to pN. The force is measured by recording the bending of the cantilever, either as a change in electrical conduction or more commonly, by monitoring a light beam reflected from the back of the cantilever into a photodetector [1, p. 183] as shown in Fig. 2.12. By adjusting the tip-sample height to keep a constant normal force while scanning across the surface, the AFM can be used to produce a surface topography map. However, when scanning perpendicularly to the cantilever axis, the frictional force can be measured as the torsion of the cantilever. By utilizing a photodetector with four quadrants (as depicted in Fig. 2.12), the normal force and friction force can be simultaneously measured as the probes scan across the surface. AFM can also be utilized to drag a nanoflake across the substrate, as demonstrated by Dienwiebel et al. [66], who attached a graphene flake to a AFM tip and dragged it across graphite. However, it should be noted that this method

concentrates the normal loading to a single point on the flake, rather than achieving an evenly distributed load.

2.4.5.2 Surface Force Apparatus

Another method worth mentioning is the Surface Force Apparatus (SFA), which consists of two curved, molecularly smooth surfaces brought into contact [1, p. 188]. The material of choice is usually mica since it can be easily cleaved into atomically flat surfaces over macroscopic areas. The sample is then placed between the two surfaces as a lubricant film, and the friction properties can be studied by applying a tangential force to the surfaces. This method provides a uniform load distribution on the surface as opposed to the setup of dragging a nanoflake by an AFM tip.

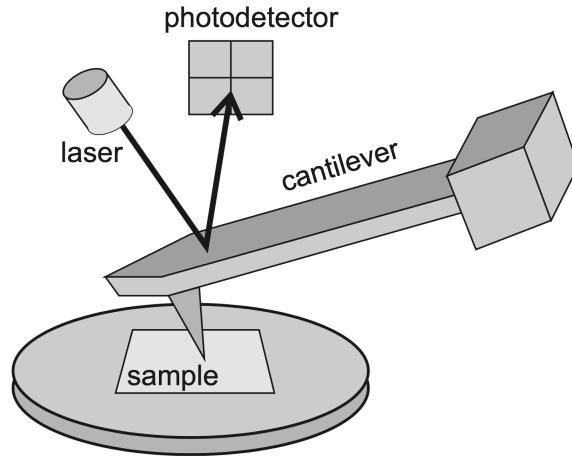


Figure 2.12: Schematic diagram of a beam-deflection atomic force microscope. Figure and caption reproduced from [1, p. 184].

2.5 Summary of previous results

Several studies have investigated the frictional behavior of graphene by varying different parameters such as normal force, sliding velocity, temperature, commensurability and graphene thickness [74]. In general, we find three types of relevant systems being studied: 1) An FFM-type setup where the graphene, either resting on a substrate or suspended, is probed by an AFM tip scanning across the surface. 2) A SFA setup with the graphene “sandwiched” in between two substrate layers moving relative to each other using the graphene as a solid lubricant. 3) A graphene flake sliding on a substrate, either being dragged by an AFM tip or by a more complex arrangement in numerical simulations. Considering that even the sharpest AFM tip will effectively put multiple atoms in contact with the sample, all methods are relatable to the study of nanoscale surface contact. However, the FFM-type is more closely related to asperity theory since it is expected to deform under increasing load, while the latter two are more aligned with the Frenkel-Kontorova type models and our specific system of interest. Nonetheless, we will consider results across all three types of systems. The most relevant studies considered are summarized in Table 2.1 for convenience.

One of the earliest tribological simulations of graphene was carried out by Bonelli et al. [82] in 2009 using a tight-binding³ method (excluding thermal excitations) to simulate a graphene flake on an infinite graphene sheet [74]. They implemented a Frenkel-Kontorova-Tomlinson-like setup where each atom in the flake is coupled horizontally to a rigid support by elastic springs. They recovered the stick-slip behavior, which is also observed in FFM setups both experimentally [75, 77] and numerically [79, 85]. Moreover, they found an agreement with the qualitative observation that soft springs allow for a clean stick-slip motion while hard springs ($\sim 40 \text{ N/m}$) inhibited it. This also aligns with the predictions of the Prandtl-Tomlinson and Frenkel-Kontorova models. In AFM and SFA experiments, the stick-slip motion tends to transition into smooth sliding when the speed exceeds $\sim 1 \mu\text{s}$ while in MD modeling the same transition is observed in the $\sim 1 \text{ m/s}$ region [24]. More precisely Liu et al. [84] finds this transition in MD simulations at 15 m/s. This 6-order-of-magnitude discrepancy has been largely discussed in connection to simplifying assumptions in MD

³The tight-binding method involves computing the electronic structure of the system, but it uses a semi-empirical approach to reduce the computational cost of the calculations. Thus, this method lies between traditional MD and more expensive ab initio methods [87].

Table 2.1: A summary of the most relevant studies considered for the review of previous results in Sec. 2.5. The table provides a distinction between the different systems being studied: FFM, SFA or flake on a substrate, as well as whether they were carried out numerically (num.) or experimentally (exp.).

| System | Type | Year | Researcher | Materials | Keywords |
|--------|------|-----------|--------------------------|---|---|
| FFM | Exp. | 2007 [75] | Zhao et al. | Si ₃ N ₄ tip on graphite. | Temperature dependence. |
| | | 2015 [76] | G. Paolicelli et al. | Si tip, graphene on SiO ₂ and Ni(111) substrate. | Load, environment, layer thickness. |
| | Num. | 2019 [77] | Zhang et al. | Monolayer graphene. | Straining of the sheet. |
| | | 2015 [78] | Yoon et al. | Si tip, graphene on SiO ₂ . | Stick-slip: tip size, scan angle, layer thickness, substrate flexibility. |
| | | 2016 [79] | Li et al. | Si tip, graphene on amorph-Si substrate. | Layer thickness, friction strengthening, stick-slip. |
| SFA | Num. | 2011 [80] | Wijn et al. | Graphene flakes between graphite. | Commensurability, rotational dynamics, superlubricity, temperature. |
| | | 2012 [37] | H. J. Kim and D. E. Kim. | Carbon sheet and nanotubes. | Corrugated nano-structured surfaces. |
| | | 2005 [66] | Dienwiebel et al. | Graphene on graphite | Commensurability, superlubricity, load. |
| Flake | Exp. | 2013 [81] | Feng et al. | Graphene on graphite. | Commensurability, superlubricity, temperature. |
| | | 2009 [82] | Bonelli et al. | Graphene on graphite. | Tight-binding, commensurability, load, flake size. |
| | Num. | 2012 [83] | Reguzzoni et al. | Graphene on graphite. | Layer thickness. |
| | | 2014 [84] | Liu et al. | Graphene on graphite. | High speed, superlubricity, rotational dynamics, sheet strain. |
| | | 2018 [85] | P. Zhu and Li | Graphene on gold. | Stick-slip, commensurability, flake size and shape. |
| | | 2019 [86] | Zhang et al. | Graphene on diamond. | Temperature, commensurability, friction coefficient. |

simulations. On the other hand, the Prandtl–Tomlinson model qualitatively disagrees as it predicts smooth sliding for low speeds only. However, in an extension of the Prandtl–Tomlinson for the study of nanoscale rolling friction by Sircar and Patra [88], they found smooth sliding for high speeds as well.

Bonelli et al. [82] also found that commensurability, through orientation of the flake and the direction of sliding, had a great impact on the frictional behavior which generally aligns with the predictions of the Frenkel-Kontorova models. They confirmed qualitatively the observation of superlubricity for certain incommensurable orientations which has been reported in experiments by Dienwiebel et al.[66] and further supported by experimental measurements of interaction energies by Feng et al. [81]. The importance of commensurability is also reported for MD simulations [80, 85, 86]. Bonelli et al. found the friction force and coefficient to be one order of magnitude higher than that of the experimental results which they attribute to the details of the numerical modeling. Generally, the experimental coefficients between graphite and most materials lie in the range of 0.08–0.18 [66]. While Dienwiebel et al. [66] reported a wide range of frictional forces from 28 ± 16 pN to 453 ± 16 pN with loads $\sim [-10, 20]$ nN, the change in friction with applied load was as low as 0.05–0.4% for the incommensurable orientations. When using the slope definition for the frictional coefficient Eq. (2.2b), this corresponds to a coefficient in the range of 0.0005–0.004. Bonelli et al. attribute the low dependency to a lacking change in contact area as the flake is loaded.

Furthermore, Bonelli et al. [82] found friction to decrease with increasing flake size which is also reported in MD simulations for graphene on gold [85]. Bonelli et al. mainly attribute this to boundary effects, but also note that the coupling to the support in their simulations made for decreased rotational freedom as flake size was increased. Thus, they hypothesized that the decreased freedom led to the graphene taking a more forced



path which is associated with a decreased stick-slip behavior. However, the general observation disagrees with the Frenkel-Kontorova models which predict the reverse; an increase in friction with increasing size.

An additional numerical study of monolayer islands of Kr on Cu by Reguzzoni and Righi [89] supports the importance of commensurability regarding size effects. They report that the effective commensurability increases drastically below a critical flake radius on the order of 10 Å. In a numerical study by Varini et al. [90], based on Kr islands adsorbed on Pb(111), this is further elaborated as they found that finite size effects are especially important for static friction due to a pinning barrier arising from the edge, preventing otherwise superlubricity due to incommensurability. They reported a relationship $F_s \sim A^{\gamma_s}$ not only sublinear, $\gamma_s < 1$, but also sublinear with respect to the island perimeter, $P \propto A^{1/2}$, by having $\gamma_s = 0.25$ for a hexagonal edge and $\gamma_s = 0.37$ when circular, indicating that only a subset of the edge is responsible for the pinning effect. This aligns with the general change in friction found by Zhu and Li [85] for different flake geometries (square, triangle, circle). Additionally, Varini et al. found the edge pinning effect to decrease with increasing temperature as the edge energy barriers are reduced. Bringing all this together, the main picture forming is that flake size, which can be related to contact area, is affecting friction through a commensurability mechanism. If the flake is constrained in some way we might not observe the same dependency. While flake size nor contact area is easily measured in experimental FFM, Mo et al. [34] found in an MD simulation that friction is proportional to contact area for an indenting sphere on a nanoscale.

Evolution effects, or so-called friction strengthening, are also found. This means that friction increases during the initial stick-slip cycles, which is observed experimentally by Zhang et al. [77] and numerically by Li et al. [79]. However, this is only found when having the graphene sheet resting on a substrate [77], as opposed to a suspended sheet. It is also found to diminish with an increasing number of graphene layers stacked (graphite) [79]. Multiple studies report a general decrease in friction with an increasing number of layers [76, 78, 79, 91, 92], but the opposite trend has also been reported [83].

A few numerical studies have investigated friction under mechanical deformations. Zhang et al. [77] found that straining a suspended graphene sheet will lower the kinetic friction. They attribute this to a modulation of flexibility which consequently changes the local pinning capability of the contact interface. Liu et al. [84] carried out an MD simulation of high-speed ballistic nanofriction (400 m/s) of graphene on graphite. They found that a biaxial stretching of the graphite substrate could be used to suppress frictional scattering and achieve persistent superlubricity. Another surface manipulating study was performed by H. J. Kim and D. E. Kim [37] who investigated the effects of corrugated nano-structured surfaces. The study revealed that the corrugation of the surfaces, involving an altering of the contact areas and structural stiffness, could result in both increased or slightly decreased friction under certain load ranges. Altogether, these studies highlight the importance of surface structure and mechanical conditions.

The friction dependency of normal load turns out to be a complex matter and has proven to be a highly system-dependent feature. As already mentioned, asperity theory mainly points to a sublinear relationship between friction and load, while the reduced-models point to a more intricate relationship through the change of the effective substrate potential which leads to an altering of the commensurability and the phonon dynamics. Experimentally rather different trends have been observed, although the majority agree on increasing friction with increasing load [1, p. 200]. For the graphene flake, Dienwiebel et al. [66] found a seemingly non-dependent relationship while a FFM study by G. Paolicelli et al. [76] yielded a sublinear relationship matching the predictions of Maugis-Dugdale theory ($F_{\text{fric}} \propto (F_N - F_{N,0})^{2/3}$). This discrepancy might be attributed to the difference in system type; a spherical tip indenting the graphene sheet as opposed to the atomic flatness of the graphene-graphite interface, which does not make for a changing contact area under load. However, numerical studies using a graphene-graphite interface still find both sublinear [82] and linear [77, 86] load dependencies.

In an experimental FFM study by Deng et al. [15] it was discovered that the friction force kept increasing after unloading the probe tip from the graphite surface. This has been argued to be a general phenomenon related to hysteresis in the adhesive interaction between two sliding bodies [93]. Following the slope definition for the friction coefficient, these results correspond to a negative friction coefficient. More recently, a negative friction coefficient has also been observed for the loading phase by Liu et al. [16] in an experimental study of the interface between graphite and muscovite mica heterojunction. With supporting numerical modeling this is attributed to “synergetic and nontrivial redistribution of water molecules at the interface”. Similar results are also reported numerically by Mandelli et al. [17] for graphite in contact with hexagonal boron nitride heterojunctions which is attributed to “load-induced suppression of the moiré superstructure out-of-plane distortions leading to a less dissipative interfacial dynamics”. Thus, the concept of a negative friction coefficient has been proven for the unloading phase of adhesive contacts and in the loading phase for a few specific systems.

The dependency of velocity is generally found to increase logarithmically with velocity in experimental AFM studies [1, p. 201] which match the low-velocity regime of the Prandtl–Tomlinson type models. At

higher velocities, thermally activated processes are less important and friction becomes independent of velocity according to the friction saturation of the Prandtl–Tomlinson model Eq. (2.7) when ignoring the athermal regime. Saturation of the velocity dependency has been observed numerically for Si tips interacting with diamond, graphite and amorphous carbon surfaces respectively with scan velocities above $1\ \mu\text{s}$ [94]. However, when considering the effects of damping the Prandtl–Tomlinson model predicts an athermal regime with viscous friction, i.e. friction being proportional to sliding velocity. Guerra et al. [95], studying gold clusters on graphite using MD simulations, found a viscous friction response in both low and high speed domains. In addition, thermal effects reversed as they found friction to decrease with increasing temperature at low speed (diffusive regime) but found friction to increase with temperature at high speed (ballistic regime). This crossover from the diffusive to the ballistic regime occurred between 1 and 10 m/s.

Regarding temperature, the general experimental trend is decreasing friction with increasing temperature as found by Zhao et al. [75] in a series of AFM graphene on graphite experiments yielding $F_{\text{fric}} \propto \exp(1/T)$. This agrees with the dominant term in the thermal drift regime of the Prandtl–Tomlinson model even though the exact temperature range does not agree. Moreover, Wijn et al. [80] found that friction commensurability can be lost at higher temperatures (above 200 K) where they found a power law behavior $F_{\text{fric}} \propto T^{-1.13 \pm 0.04}$. Numerically, Zhang et al. [86] found that friction increased with temperature, using a sliding speed of 10 m/s. Considering the findings of Guerra et al. [95] this qualitative different behavior can be attributed to the transition from low speed diffusive friction to high speed ballistic friction in the MD simulations.

From the review of previous results, we find several gaps and discrepancies in the description of friction provided by the reduced-models, MD simulations and experimental methods respectively. Some of the discrepancies can be attributed to the fact that different physical mechanisms are included in the numerical modeling. The reduced-models provide a simplistic description, while the MD simulations are expected to capture a more complex behavior. We might also point to differences in the studied systems as an important factor to consider. This includes the physical conditions such as sliding speed and temperature, but also higher-level features related to the mechanical properties of the system. For instance, the FFM-based results consider an asperity-like system where the tip is expected to deform under loading, which gives rise to a change in the contact area. This feature is lacking for the flake on a substrate, and thus we might question the role of the contact area in these systems. More precisely, when inflicting an out-of-plane buckling through Kirigami cuts and stretching, the contact area is expected to decrease as well for which asperity theory predicts a decrease in friction. However, as the system undergoes deformation, it may also lead to a change in commensurability, which can result in significant modifications to the friction due to its effect on stick-slip behavior. Based on the results obtained from a non-cut sheet under tension, there are indications that strain alone can lead to a reduction in friction, even without taking into account the contact area. Similarly, the findings from a corrugated nanosurface suggest that surface stiffness may also be a significant factor in determining friction for our nanoscale Kirigami system.

2.6 Research questions

Based on the review of friction presented in Chapter 2, it is evident that the behavior of friction is influenced by various factors, such as the specific system under investigation, the numerical modeling approach, and the physical conditions related to the environment and the probing of friction. In our study, we aim to investigate the frictional behavior of a Kirigami sheet under the effects of strain. Previous studies have demonstrated that strained Kirigami sheets are prone to out-of-plane buckling [6, 7] which is indicative of a possible transition between two distinct systems: An atomically flat interface and an asperity system. These systems are usually only studied separately, and therefore, our primary objective is to investigate the possible frictional effects linked to strain-induced system transformations. In particular, we want to investigate the significance of the contact area and evaluate the hypothesis that reducing the contact area will lead to a decrease in friction. Additionally, we seek to examine the relationship between the friction-load curve and this phenomenon. For the sake of contributing new insight to the field of nanoscale friction, we are interested in non-linear dependencies between friction and strain for various Kirigami designs. Drawing on this perspective, we aim to investigate the prospects of achieving a negative friction coefficient for a system of coupled load and strain. In order to contextualize our findings within the theoretical framework, we will take into account the results from prior studies.

To gain a more comprehensive understanding of the potential applications of Kirigami design, we aim to develop a dataset based on MD simulations that capture the frictional effects on Kirigami designs when subjected to strain and load. We intend to employ machine learning techniques to discern any meaningful trends in the data that may be used to inform future research endeavors. Specifically, we seek to leverage the machine learning model to facilitate an accelerated search for optimizing specific frictional properties.



Our focus will be on evaluating the prospects of reducing or increasing the friction force, as well as reducing or increasing the friction coefficient for a coupled system of load and strain. Our main research questions can be summarized as follows.

1. How can we design an MD simulation that provides a reliable foundation for an investigation of the frictional behavior for a Kirigami graphene sheet sliding on a substrate? How do physical conditions such as temperature and sliding speed control friction?
2. How can we design an ensemble of Kirigami patterns for the investigation of its frictional properties with the scope of getting out-of-plane buckling and also randomized design features?
3. Can we control friction for a Kirigami sheet through Kirigami pattern design and straining of the sheet?
 - (a) Does friction dependent on a changing contact area?
 - (b) How does the friction-load curve relate to strained Kirigami sheets?
 - (c) Are the effects of strain and pattern design significant when considered independently?
 - (d) Is the frictional behavior consistent with the Prandtl–Tomlinson, Frenkel-Kontorova and Frenkel-Kontorova-Tomlinson models?
4. Is it possible to utilize machine learning to identify general trends in the relationship between friction and kirigami patterns, strain and load?
5. Can we use a trained machine learning model to predict new designs through an accelerated search?
6. What are the prospects of achieving a negative friction coefficient for a system of coupled load and strain through Kirigami design?

Chapter 3

Molecular Dynamics

Molecular Dynamics (MD) is an atomistic simulation method that is commonly employed in the investigation of atomic-scale friction due to its ability to track each atom in a system [49]. In recent years, advances in computing algorithms and hardware have made MD simulations increasingly capable of simulating tribological systems [24]. We will utilize MD as our primary numerical approach to examine the frictional behavior of a nanoscale Kirigami sheet sliding on a substrate. The small-scale modifications associated with nanoscale Kirigami are still beyond the reach of experimental approaches, and the complexity of the system precludes analytical solutions as well. Hence, MD simulations represent one of the few viable options for addressing this problem.

An MD simulation can be viewed as a “computational experiment”, where we specify a set of initial conditions and evolve the system to measure certain properties of interest. This is done through the definition of interatomic force fields which allow us to solve Newton’s equations of motion by numerical integration [73, p. 303]. Other more sophisticated and accurate approaches exist, like ab initio MD which utilizes electronic structure calculations at simulation time [96]. One of the most popular methods is based on density functional theory (DFT) [97] which considers quantum mechanical modeling of the electronic state of the system. However, such methods are rarely used in sliding friction simulations since the computational cost makes it only feasible to handle relatively small systems, typically hundreds of atoms, for relatively short durations, typically much less than 1 ns [61]. In addition, we aim to perform multiple simulations under the change of various physical parameters which adds extra demands on the computational resources.

In this chapter, we introduce the fundamental principles of MD modeling and describe our implementation choices for the system of interest. We will focus on the key aspects related to our implementation rather than providing a comprehensive analysis of all available techniques.

3.1 Potentials

The interatomic force fields governing the MD simulation are given from the choice of potentials and have a significant impact on the outcomes obtained. The potentials can vary from intricate energy surfaces that consider electrons at either the density-functional or tight-binding level to angle-dependent many-particle potentials, basic pairwise potentials, or simple models of elastic springs, and extensions of Frenkel-Kontorova-type formulations [61]. For the choice of potentials, and materials, we take a basis in the numerical MD study by Li et al. [79] simulating a FFM type setup where a silicon tip indents a graphene sheet supported by a silicon substrate. Our system obviously differs from this arrangement since we will be sliding the entire sheet upon the substrate. Nevertheless, we contend that this serves as an appropriate basis for selecting the potentials based on the materials involved. Thus, we adopt the potentials from [79] describing the covalent bonds between carbon atoms (C–C) in the graphene sheet with the Tersoff potential [98] and the covalent bonds between silicon atoms (Si–Si) in the substrate with the Stillinger–Weber potential [99]. A typical 12–6 Lennard–Jones (LJ) potential is used to describe the van der Waals adhesive interaction between the graphene sheet and the substrate.

3.1.1 General formulation of potentials

The potentials determine the interatomic forces in the MD simulation, with the force \mathbf{F} acting on an atom being derived from the potential energy U as the derivative $\mathbf{F} = -\nabla U$, where $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})$. The energy



of N interacting particles can be described as an expansion in terms of participating particles as

$$U = \sum_i V_1(\mathbf{r}_i) + \sum_{i,j} V_2(\mathbf{r}_i, \mathbf{r}_j) + \sum_{i,j,k} V_3(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k) + \dots,$$

where \mathbf{r}_n is the position of the n^{th} particle and V_m is called an m -body potential [98]. The first one-body term corresponds to an external potential (e.g. gravity), followed by the two-body term, the three-body term and so on. The simplest model that includes particle interaction is the pair potential truncating the expansion after the two-body term. A general feature of the pair potentials is that they favor close-packed structures that are unsuited to describe covalent bonds which take more open structures. In particular, pair potentials are completely inapplicable to strongly covalent systems [98]. In order to accommodate the description of covalent bonds, we include the three-body term in both the Stillinger–Weber and Tersoff potential. For the interaction between the sheet and the substrate, we use the LJ pair potential describing the non-bonded van der Waals interaction which is often used to treat interactions between surfaces in friction simulations [3, 78, 85, 100]. In the following sections Sec. 3.1.2 to 3.1.4 we will introduce each of the potentials in more detail.

3.1.2 Lennard Jones

The theoretical basis in this section is based on [101–103]. The Lennard-Jones (LJ) model is one of the most commonly used pair potentials for MD simulations. LJ models the potential energy between two non-bonding atoms solely based on interatomic distance. The model accounts for long-ranged attractive forces arising from London dispersion forces (dipole-induced dipole) and repulsive forces that capture the hard core of overlapping electron orbitals at small distances (Pauli repulsion). Thus, it assumes neutrally charged atoms and was originally proposed for noble gases. The classical 12–6 version of the model, referring to the powers of the repulsive and attractive forces respectively, reads

$$U = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right], \quad r < r_c, \quad (3.1)$$

where r is the interatomic distance with cut-off r_c , ϵ is the depth of the potential well and σ the interatomic distance where the potential is zero. The potential is illustrated in Fig. 3.1. By solving for the potential minimum ($dU/dr = 0$) we find the equilibrium distance to be $r_0 = \sigma^{2/3}$. This makes for a slightly more intuitive interpretation of σ as the parameter which sets the equilibrium distance between atoms, i.e. the dividing line for which the force is repulsive or attractive. We will adopt the potential parameters from Li et al. [79] with $\sigma = 3.0 \text{ \AA}$ and $\epsilon = 0.0092 \text{ eV}$.

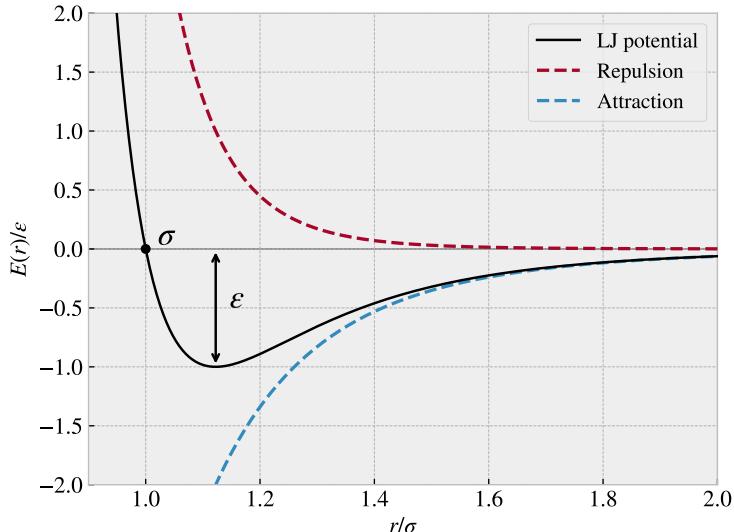


Figure 3.1: Illustration of the LJ potential described by Eq. (3.1) showing the contributions from the repulsive and attractive part of the potential.

3.1.3 Stillinger-Weber

The theoretical background of this section is based on [99, 104]. The Stillinger-Weber potential takes the form of a three-body potential

$$U = \sum_i \sum_{j>i} \phi_2(r_{ij}) + \sum_i \sum_{j\neq i} \sum_{k>j} \phi_3(r_{ij}, r_{ik}, \theta_{ijk}),$$

where r_{ij} denotes the distance between atom i and j , and θ_{ijk} the angle between bond ij and jk (see Fig. 3.2). The first sums run over all pair interactions while the last sums run over triplets of particles, for neighbors j and k of atom i within a cut-off distance $r = a\sigma$.

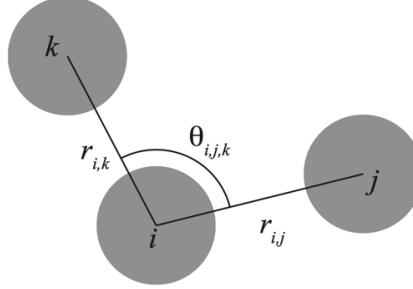


Figure 3.2: Illustration showing the definition of the angle θ_{ijk} .

The two-body term ϕ_2 builds from the LJ model with the addition of an exponential cutoff term

$$\phi_2(r_{ij}) = A_{ij}\epsilon_{ij} \left[B_{ij} \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{p_{ij}} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{q_{ij}} \right] \exp \left(\frac{\sigma_{ij}}{r_{ij} - a_{ij}\sigma_{ij}} \right). \quad (3.2)$$

The model parameters A , ϵ , B , σ , p , q and a come with i, j indices to indicate that these parameters should be specified for each unique pair of atom types. However, in our case, we will only provide a single value for each model parameter since we are exclusively dealing with Si–Si bonds. We see that the first term in Eq. (3.2) is reminiscent of the LJ model in Eq. (3.1) while the last term effectively drives the potential to zero at $r = a\sigma$, which is the chosen cut-off distance for the potential evaluation. With the chosen model parameters for the Si–Si bonds (see Table 3.1), the cut-off becomes ~ 3.8 Å. The three body term includes an angle dependency as

$$\phi_3(r_{ij}, r_{ik}, \theta_{ijk}) = \lambda_{ijk} \epsilon_{ijk} \left[\cos \theta_{ijk} - \cos \theta_{0,ijk} \right]^2 \exp \left(\frac{\gamma_{ij}\sigma_{ij}}{r_{ij} - a_{ij}\sigma_{ij}} \right) \exp \left(\frac{\gamma_{ik}\sigma_{ik}}{r_{ik} - a_{ik}\sigma_{ik}} \right), \quad (3.3)$$

where $\theta_{0,ijk}$ is the equilibrium angle. The first term of Eq. (3.3) includes an angle dependency analog to a harmonic oscillator based on a cosine angle distance from the equilibrium angle. The final two terms act again as a cut-off function by driving the potential to zero at $r_{ij} = a_{ij}\sigma_{ij}$ and $r_{ik} = a_{ik}\sigma_{ik}$ respectively. We adopt the parameters for the modeling of the Si–Si bonds suggested in the original paper by Stillinger and Weber [99] which is shown in Table 3.1 along with an interpretation of each model parameter.



Table 3.1: Parameters for the Stillinger-Weber potential used for the modeling of the Si–Si bonds in the silicon substrate. The parameters are adopted from [99].

| Parameter | Value | Description |
|------------------|--------------|---|
| ϵ | 2.1683 eV | Depth of the potential well for each pair and triplets of atoms. |
| σ | 2.0951 Å | Distance for which the individual pair interactions has zero potential (analog to the LJ model). |
| a | 1.80 | The cut-off distance for each pair of atoms in units of σ . |
| λ | 21.0 | The overall depth of the three-body potential well. |
| γ | 1.20 | Determines the shape of the three-body cut-off terms. |
| $\cos(\theta_0)$ | -1/3 | Cosine of the equilibrium angle. |
| A | 7.049556277 | The overall depth of the two-body potential well. |
| B | 0.6022245584 | Scales the repulsion part of the two-body term. |
| p | 4.0 | The power dependency for the repulsion part of the two-body term. |
| q | 0.0 | The power dependency for the attraction part of the two-body term. |
| tol | 0 | (LAMMPS specific) Option to define a different cut-off than the theoretical $r = a\sigma$. tol = 0 refers to the use of the theoretical cut-off. |

3.1.4 Tersoff

The theoretical basis in this section is based on [98, 105]. The Tersoff potential abandons the idea of a general m -body form and attempts instead to build the model on a more physics-informed approach; the more neighbors an atom has the weaker the bonds will be. Thus, it introduces the bond order (bond strength), which is environment specific and decreases with increasing bond coordination (number of neighbors for a given atom). A sketch of the Tersoff potential can be seen in Fig. 3.3.

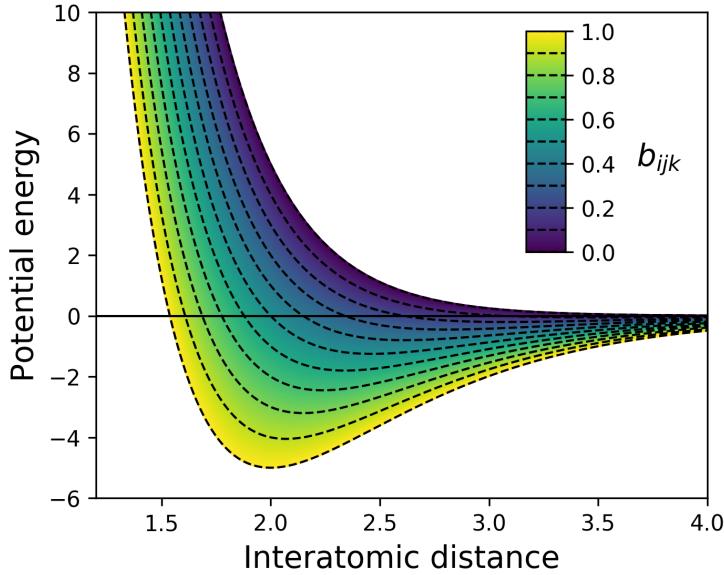


Figure 3.3: Sketch of the potential energy for a Tersoff-type potential. The energy minimum is shifted with changing bond order b_{ijk} . Reproduced from [106].

The potential energy is taken to have the form

$$U = \sum_i U_i = \frac{1}{2} \sum_{i \neq j} V_{ij},$$

$$V_{ij} = f_C(r_{ij}) [f_R(r_{ij}) + b_{ij} f_A(r_{ij})],$$

where the total potential energy U is decomposed into a bond energy V_{ij} . The indices i and j run over the atoms of the system with r_{ij} denoting the distance between atom i and j . Notice that the sum includes all

combinations of i, j but $i \neq j$, meaning that an atom cannot bond to itself. However, we count other bonds twice, e.g. (1, 2) and (2, 1), which is the explanation for the additional factor 1/2. The reasoning for the double counting lies in the asymmetry of the bond order $b_{ij} \neq b_{ji}$ leading to $V_{ij} \neq V_{ji}$. The bond energy is composed of a repulsive term f_R , arising from overlapping wave functions, and an attractive term f_A associated with bonding. f_C is simply a smooth cut-off function to increase computational efficiency. b_{ij} represent the bond order, i.e. the strength of the bonds, which depends inversely on the number of bonds, the bond angles (θ_{ijk}) and optionally the relative bond lengths (r_{ij}, r_{jk}). Notice that an additional cut-off term a_{ij} was originally multiplied to f_R as a way of limiting the range of the repulsive interactions to the first neighbor shell. This is similar to the role of b_{ij} for the attractive term f_A , but it is often omitted for the repulsive term f_R , and we do so as well by setting $a_{ij} = 1$.

The cut-off function f_C goes from 1 to 0 over a small interval range $R \pm D$ as

$$f_C(r) = \begin{cases} 1 & r < R - D \\ \frac{1}{2} - \frac{1}{2} \sin\left(\frac{\pi}{2} \frac{r-R}{D}\right) & R - D < r < R + D \\ 0 & r > R + D \end{cases}$$

which is continuous and differentiable for all r . R is usually chosen to include only the first neighbor shell. The repulsive and attractive terms f_R and f_A are modeled as an exponential function, similar to a Morse potential,

$$\begin{aligned} f_R(r) &= A \exp(-\lambda_1 r), \\ f_A(r) &= -B \exp(-\lambda_2 r). \end{aligned}$$

The novel feature of the Tersoff model lies in the modeling of the bond order b_{ij} which includes three-body interactions by summing over a third atom $k \neq i, j$ within the cut-off $r_{ik} < R + D$ as shown in the following.

$$b_{ij} = (1 + \beta^n \zeta_{ij}^n)^{-\frac{1}{2n}} \quad (3.4)$$

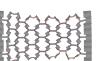
$$\zeta_{ij} = \sum_{k \neq i, j} f_C(r_{ik}) g\left(\theta_{ijk}(r_{ij}, r_{ik})\right) \exp\left(\lambda_3^m (r_{ij} - r_{ik})^m\right) \quad (3.5)$$

$$g(\theta) = \gamma_{ijk} \left(1 + \frac{c^2}{d^2} - \frac{c^2}{[d^2 + (\cos \theta - \cos \theta_0)^2]} \right). \quad (3.6)$$

In Eq. (3.6) ζ_{ij} is an effective coordination and $g(\theta)$ captures angle dependency and is minimized at the equilibrium angle $\theta = \theta_0$. The parameters used to model the graphene C–C bonds are adopted from J. Tersoff [107] and summarized in Table 3.2.

Table 3.2: Parameters for the Tersoff potential used for the modeling of the C–C bonds in the graphene sheet. The parameters are adopted from [107].

| Parameter | Value | Description |
|------------------|-------------------------|--|
| R | 1.95 Å | Center distance for cut-off. |
| D | 0.15 Å | Thickness of cut-off region. |
| λ_1 | 3.4879 Å ⁻¹ | Decay of repulsion potential term f_R . |
| λ_2 | 2.2119 Å ⁻¹ | Decay of attractive potential term f_A . |
| A | 1393.6 eV | Repulsion potential maximum at the core ($f_R(r_{ij} = 0)$). |
| B | 346.74 eV | Attractive potential minimum at core ($f_A(r_{ij} = 0)$). |
| β | 1.5724×10^{-7} | Base for the exponential scaling of the effective coordination affecting the bond strength b_{ij} . |
| n | 0.72751 | Power law exponent for the bond order dependency. |
| λ_3 | 0.0 Å ⁻¹ | Base for the exponential cut-off of the effective coordination ζ_{ij} . |
| m | — | Exponent for the exponential cut-off of the effective coordination ζ_{ij} . Not relevant since $\lambda_3 = 0$. |
| γ | 1.0 | Linear scaling of the angle dependency term. |
| c | 3.8049×10^4 | Strength of the angular effect. |
| d | 4.3484 | Determines the “sharpness” of the angular dependency. |
| $\cos(\theta_0)$ | -0.57058 | Cosine of the equilibrium angle. |



3.2 Integration

Assuming that one has defined a system of atoms, defining the atom types, initial positions and velocities, and interatomic potentials, we need to move the system forward in time. By solving Newton's equations of motion we achieve this by effectively sampling the microcanonical ensemble characterized by a constant number of particles N , volume V and energy E , hence denoted NVE [108]. Newton's equations of motion read

$$m_i \frac{d^2 \mathbf{r}_i}{dt^2} = \mathbf{F}_i = -\nabla U_i, \quad (3.7)$$

where i is the atom index, m_i its mass, $\mathbf{r}_i = (x_i, y_i, z_i)$ the position, t is time, $\nabla_i = (\frac{\partial}{\partial x_i}, \frac{\partial}{\partial y_i}, \frac{\partial}{\partial z_i})$ and U_i the potential energy. The potential energy is defined by the interatomic potentials and any external forces applied to the system. Since the forces defined by the potentials are conservative we expect the energy of the solution to be conserved. We can redefine Eq. (3.7) in terms of two coupled first order differential equations

$$\dot{\mathbf{v}}_i(t) = \frac{\mathbf{F}}{m_i}, \quad \dot{\mathbf{r}}_i(t) = \mathbf{v}_i(t), \quad (3.8)$$

where $\dot{x} = dx/dt$ is Newton's notation for the time derivative and $\mathbf{v} = (v_x, v_y, v_z)$ is velocity. Numerically we can solve the coupled equations by integrating over discrete timesteps. That is, we discretize the solution into temporal steps $t_k = t_0 + k\Delta t$, $k = 0, 1, \dots, (T - t_0)/\Delta t$ with start time t_0 , timestep Δt and total time T . The choice of timestep should be chosen small enough to avoid instabilities in the numerical solution.

3.2.1 Velocity Verlet

A popular approach to the numerical integration of Newton's equations of motion, when written as two coupled first-order differential equations Eq. (3.8), is the *velocity verlet* algorithm. We can derive the algorithm by the use of Taylor expansions. We begin by expanding the next-step position vector $\mathbf{r}_i(t + \Delta t)$ at time t

$$\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i(t) + \dot{\mathbf{r}}_i(t)\Delta t + \frac{\ddot{\mathbf{r}}_i(t)}{2}\Delta t^2 + \mathcal{O}(\Delta t^3), \quad (3.9)$$

where $\ddot{\mathbf{r}} = d^2\mathbf{r}/dt^2$ and Δt^n is simply the relaxed notation for $(\Delta t)^n$. The remaining term $\mathcal{O}(\Delta t^3)$ is big O notation for the truncation including a dependence of Δt^3 and higher order. Similarly, we take the expansions of the next-step velocity vector $\mathbf{v}_i(t + \Delta t)$ at time t

$$\mathbf{v}_i(t + \Delta t) = \mathbf{v}_i(t) + \dot{\mathbf{v}}_i(t)\Delta t + \frac{\ddot{\mathbf{v}}_i(t)}{2}\Delta t^2 + \mathcal{O}(\Delta t^3). \quad (3.10)$$

Finally, by taking the expansion of $\dot{\mathbf{v}}_i(t + \Delta t)$ we can eliminate the $\ddot{\mathbf{v}}_i$ -term in Eq. (3.10) and simplify it as shown in the following.

$$\begin{aligned} \dot{\mathbf{v}}_i(t + \Delta t) &= \dot{\mathbf{v}}_i(t) + \ddot{\mathbf{v}}_i(t)\Delta t + \mathcal{O}(\Delta t^2) \\ \frac{\ddot{\mathbf{v}}_i(t)}{2}\Delta t^2 &= \frac{\Delta t}{2} \left(\dot{\mathbf{v}}_i(t + \Delta t) - \dot{\mathbf{v}}_i(t) \right) + \mathcal{O}(\Delta t^3) \\ &\Downarrow \\ \mathbf{v}_i(t + \Delta t) &= \mathbf{v}_i(t) + \dot{\mathbf{v}}_i(t)\Delta t + \frac{\Delta t}{2} \left(\dot{\mathbf{v}}_i(t + \Delta t) - \dot{\mathbf{v}}_i(t) \right) + \mathcal{O}(\Delta t^3) \\ &= \mathbf{v}_i(t) + \frac{\Delta t}{2} \left(\dot{\mathbf{v}}_i(t) + \dot{\mathbf{v}}_i(t + \Delta t) \right) + \mathcal{O}(\Delta t^3). \end{aligned} \quad (3.11)$$

By combining Eq. (3.9) and Eq. (3.11) using $\dot{\mathbf{v}} = \mathbf{F}_i(t)/m_i$ and $\mathbf{v} = \dot{\mathbf{r}}$ we arrive at the final scheme

$$\begin{aligned} \mathbf{r}_i(t + \Delta t) &= \mathbf{r}_i(t) + \mathbf{v}_i(t)\Delta t + \frac{\mathbf{F}_i(t)}{2m_i}\Delta t^2 + \mathcal{O}(\Delta t^3), \\ \mathbf{v}_i(t + \Delta t) &= \mathbf{v}_i(t) + \frac{\mathbf{F}_i(t) + \mathbf{F}_i(t + \Delta t)}{2m_i}\Delta t + \mathcal{O}(\Delta t^3). \end{aligned}$$

This scheme will give a local error on the order Δt^3 corresponding to a global error on the order Δt^2 . One of the most popular ways to implement this numerically is as stated in the following steps.

1. Calculate $v_{k+\frac{1}{2}} = v_k + \frac{F_k}{2m} \Delta t$.
2. Calculate $r_{k+1} = r_k + v_{k+\frac{1}{2}} \Delta t$.
3. Evaluate the force $F_{k+1} = F(r_{k+1})$.
4. Calculate $v_{k+1} = v_{k+\frac{1}{2}} + \frac{F_{k+1}}{2m} \Delta t$.

3.3 Thermostats

In Chapter 2 we introduced friction as an ultimate result of the equipartition theorem stating that the kinetic energy supplied by the sliding motion will tend to transfer to other degrees of freedom and eventually dissipate to the environment as heat through phonon transport (and electrons for a metallic system) [24]. However, when modeling the system exclusively through the solutions of Newton's equations of motion we have no dissipation channel in our system. Instead, the energy will reflect back and forth and eventually “pile up” in the system in an unphysical manner. In order to resolve this issue we have to model the heat dissipation to the environment. This can be approached in a variety of ways, but one of the more common choices, which we will use as well, is the Langevin thermostat.

3.3.1 Langevin thermostat

The Langevin thermostat is a stochastic thermostat that modifies Newton's equations of motion such that the solution lies in the canonical ensemble characterized by a constant number of particles N , constant volume V and constant temperature T , hence denoted NVT [24]. When going from the microcanonical ensemble NVE , described by Newton's equations of motion Eq. (3.7), to the canonical ensemble NVT , we effectively perform a Legendre transformation which substitutes temperature for energy in the regard to which variables are held constant. The canonical ensemble is represented by a finite system being in contact with an infinite heat bath of temperature T . The NVT ensemble is equivalent to sampling a system in thermodynamic equilibrium where the weight of each microscopic state is given by the Boltzmann factor $\exp[-E/(k_B T)]$ where k_B is the Boltzmann constant.

The Langevin thermostat is governed by the Langevin equation which originated as the modified version of Newton's equations for a Brownian particle [109]. A Brownian particle is a small particle suspended in liquid, e.g. pollen or dust, named after Robert Brown (1773–1858) who was the first to observe its jittery motion. The Langevin equation describes this motion as the combination of a viscous drag force $-\alpha \mathbf{v}$, where α is a positive friction coefficient and \mathbf{v} the velocity vector, and a random fluctuation force \mathbf{R} . The Langevin equation reads

$$m \frac{d\mathbf{v}}{dt} = -\alpha \mathbf{v} + \mathbf{R}, \quad (3.12)$$

where m is the particle mass. This effectively describes the particle of interest, the Brownian particle, as being suspended in a sea of smaller particles. The collision with these smaller particles is then modeled by the combined effects of the drag force and the fluctuation force. If the fluctuation force is excluded Eq. (3.12) becomes

$$m \frac{d\mathbf{v}}{dt} = -\alpha \mathbf{v} \quad \Rightarrow \quad \mathbf{v}_i(t) = \mathbf{v}(0) e^{-\frac{\alpha t}{m}},$$

where the solution reveals that the Brownian particle will come to a complete stop after a long time $\mathbf{v}_i(t \rightarrow \infty) \rightarrow \mathbf{0}$. This is in violation with the equipartition theorem which dictates a non-zero average squared velocity in equilibrium $\langle v^2 \rangle_{\text{eq}}$ as

$$\frac{1}{2} m \langle v^2 \rangle_{\text{eq}} = \frac{k_B T}{2}. \quad (3.13)$$

Hence, the fluctuation force is necessary to obtain the correct equilibrium. In the following, we will attempt to introduce the reasoning behind the Langevin equation using only one dimension in order to simplify the notation a bit. The theoretical basis in this section is based on [109].

We describe the statistical nature of the collisions as a sum of independent momentum transfers

$$\Delta P = \sum_i^N \delta p_i,$$



where ΔP denotes the change of momentum after N momentum transfers δp_i from the environment to the Brownian particle. We assume the first and second moments to be $\langle \delta p \rangle = 0$ and $\langle \delta p^2 \rangle = \sigma^2$. When N is large the central limit theorem states that the random variable ΔP has a gaussian distribution with $\langle P \rangle = 0$ and $\langle \Delta P^2 \rangle = N\sigma^2$. If we consider the momentum change ΔP over a discrete time Δt , where the number of collisions is proportional to time $N \propto \Delta t$, the corresponding fluctuation force $R = \Delta P / \Delta t$ will have a variance

$$\langle R^2 \rangle = \frac{\langle \Delta P^2 \rangle}{\Delta t^2} = \frac{N\sigma^2}{\Delta t^2} \propto \frac{1}{\Delta t}.$$

In an MD simulation, we pick a random force $R(t)$ from a Gaussian distribution every timestep Δt . These random forces will not be correlated as long as Δt is larger than the correlation time from the molecular fluid, which we will assume here. However, there exist corrections to this approximation, but we will not consider these for our scope. By assuming that this criterion is met, we can write the correlation function as

$$\langle R(t)R(0) \rangle = \begin{cases} \frac{a}{\Delta t}, & |t| < \Delta t/2 \\ 0, & |t| > \Delta t/2, \end{cases} \quad (3.14)$$

where the constant a describes the magnitude of the fluctuations. We could in principle determine a from the variance of ΔP , but instead, we will determine it from the equipartition principle. In the limit $\Delta t \rightarrow 0$ the correlation function becomes

$$\langle R(t)R(0) \rangle = a\delta(t), \quad (3.15)$$

where δ denotes the Dirac delta function. This is valid for all spatial coordinates which are all independent of each other. Since both the drag force and the fluctuation force originate from the molecular fluid, where the drag force $-\alpha v$ carries a velocity dependency, it is reasonable to assume that fluctuation force is independent of velocity, i.e. $\langle R_i v_j \rangle = 0$ for all cartesian indices i and j . We can justify the physical motivation for the Langevin equation by determining the relationship between the drag coefficient α and the random force R [109]. From the Langevin equation Eq. (3.12) we can compute the velocity autocorrelation function. Note that we continue to use only one dimension for simplicity. We begin by multiplying by $(e^{\alpha t/m})/m$

$$\dot{v}(t)e^{\alpha t/m} + \frac{\alpha}{m}v(t)e^{\frac{\alpha t}{m}} = \frac{F}{m}e^{\frac{\alpha t}{m}}.$$

We integrate from $t = -\infty$, using integration by parts on the latter term on the left-hand side, in order to calculate the velocity

$$\begin{aligned} \int_{-\infty}^t dt' \dot{v}(t')e^{\frac{\alpha t'}{m}} + \frac{\alpha}{m}v(t')e^{\frac{\alpha t'}{m}} &= \int_{-\infty}^t dt' e^{\frac{\alpha t'}{m}} \frac{F(t')}{m} \\ \int_{-\infty}^t dt' \dot{v}(t')e^{\frac{\alpha t'}{m}} + \left(\left[v(t')e^{\frac{\alpha t'}{m}} \right]_{-\infty}^t - \int_{-\infty}^t dt' \dot{v}(t')e^{\frac{\alpha t'}{m}} \right) &= \int_{-\infty}^t dt' e^{\frac{\alpha t'}{m}} \frac{F(t')}{m} \\ v(t) &= \int_{-\infty}^t dt' e^{-\frac{\alpha(t-t')}{m}} \frac{F(t')}{m}, \end{aligned}$$

where $e^{-\frac{\alpha t}{m}}$ plays the role of a response function. We can then calculate the autocorrelation

$$\begin{aligned} \langle v(t)v(0) \rangle &= \int_{-\infty}^t dt_1 \int_{-\infty}^0 dt_2 e^{-\frac{\alpha(t-t_1-t_2)}{m}} \frac{\langle F(t_1)F(t_2) \rangle}{m^2} \\ &= \int_{-\infty}^t dt_1 \int_{-\infty}^0 dt_2 e^{-\frac{\alpha(t-t_1-t_2)}{m}} \frac{a\delta(t_1-t_2)}{m^2} \\ &= \int_{-\infty}^0 dt_2 e^{-\frac{\alpha(t-2t_2)}{m}} \frac{a}{m^2} = \frac{a}{2m\alpha} e^{-\frac{\alpha t}{m}}, \end{aligned}$$

where we used Eq. (3.15) and the fact that the integration commutes with the average (we are allowed to flip the order). By comparing this with the equipartition theorem we get

$$\begin{aligned} \frac{1}{2}m\langle v^2 \rangle &= \frac{k_B T}{2} \\ \frac{1}{2}m\langle v(0)v(0) \rangle &= \frac{a}{4\alpha} = \frac{k_B T}{2} \\ a &= 2\alpha k_B T. \end{aligned}$$

We notice the appearance of α meaning that the magnitude of the random force fluctuations a increase both with viscous friction ($-\alpha\mathbf{v}$) and temperature T . Moreover, we can integrate the velocity over time to get displacement $x(t)$ and show that the variance is [109]

$$\langle x^2(t) \rangle = \frac{2k_B T}{\alpha} \left(t - \frac{m}{\alpha} \left(1 - e^{-\alpha t/m} \right) \right).$$

For $t \gg m/\alpha$, only the t -term will survive yielding

$$\langle x^2(t) \rangle = 2k_B T t / \alpha.$$

In 1D, the diffusion constant D is related to the variance as $\langle x^2 \rangle = 2Dt$, meaning that this represents the Einstein relation $D = \mu k_B T$ with the mobility $\mu = 1/\alpha$. When $t \ll m/\alpha$, we use the Taylor expansion $1 - e^{-x} \approx x - x^2/2$ for $x \ll 1$ which gives

$$\langle x^2(t) \rangle = \frac{k_B T}{m} t^2.$$

Using $\langle x^2 \rangle/t^2 = \langle v^2 \rangle$ we see that the result is in agreement with the equipartition theorem Eq. (3.13)

$$\langle v^2(t) \rangle = \frac{k_B T}{m} \iff \frac{1}{2} m \langle v^2 \rangle_{\text{eq}} = \frac{k_B T}{2}.$$

Thus, we find the finite damping correlation time α/m to describe the crossover between the ballistic regime $\sqrt{\langle x^2(t) \rangle} \propto t$ at $t \ll m/\alpha$ to the diffusive regime $\sqrt{\langle x^2(t) \rangle} \propto \sqrt{t}$ at $t \gg m/\alpha$. That is, at short time scales the thermal movement can be thought of as relatively free with a low collision rate, while at longer time scales the movement is characterized by a jittery random-walk-looking motion that is related to diffusion.

3.3.2 Applying the Langevin Thermostat

The numerical implementation of the Langevin equation Eq. (3.12) is done through the simulation software LAMMPS (see Sec. 3.5) following [110] by defining the force vector for each particle as

$$\begin{aligned} \mathbf{F} &= \mathbf{F}_c + \mathbf{F}_f + \mathbf{F}_r \\ &= -\nabla U - \gamma m \mathbf{v} + \sqrt{\frac{2k_B T m \gamma}{\Delta t}} \mathbf{h}(t), \end{aligned} \quad (3.16)$$

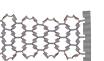
where \mathbf{F}_c is the added conservative force computed via the usual interatomic interactions described by the potential U , \mathbf{F}_f is the drag force described as a damping term $-\gamma m \mathbf{v}$ with $\alpha = \gamma m$, and \mathbf{F}_r is the random fluctuation force where \mathbf{h} is a random vector drawn from a normal distribution with zero mean and unit variance. The fact that Δt now appears in the denominator for the random force variance $2k_B T m \gamma / \Delta t$ is due to the discretization of time. By applying Eq. (3.16) we get the refined velocity verlet scheme

$$\begin{aligned} \mathbf{v}_i(t + \Delta t/2) &= \mathbf{v}_i(t) - \frac{\Delta t}{2} \left(\frac{\nabla_i U(t)}{m_i} + \gamma \mathbf{v}_i \right) + \sqrt{\frac{k_B T \gamma \Delta t}{2m_i}} \mathbf{h}_i, \\ \mathbf{r}_i(t + \Delta t) &= \mathbf{r}_i(t) + \mathbf{v}_i(t + \Delta t/2) \Delta t, \\ \mathbf{v}_i(t + \Delta t) &= \mathbf{v}_i(t + \Delta t/2) - \frac{\Delta t}{2} \left(\frac{\nabla_i U(t + \Delta t)}{m_i} + \gamma \mathbf{v}_i(t + \Delta t/2) \right) + \sqrt{\frac{k_B T \gamma \Delta t}{2m_i}} \mathbf{h}_i, \end{aligned}$$

with new random vector \mathbf{h}_i for each particle and each update. Notice however, we will only apply the thermostat to specific regions in our simulation, mainly on the outer edges, while the main part of interest is modeled exclusively by Newton's equations of motion as described in Eq. (3.7). This is done in order to avoid affecting the governing parts of the friction simulation too much. We use a damping of $1/\gamma = m/\alpha = 1 \text{ ps}$ as a common default choice.

3.4 Limitations

On a general note, MD simulations are limited to relatively small time and size scales due to the available computation time. Modern CPUs perform on the order of 10^9 floating-point operations per second (FLOPS) per core [61]. MD simulations can benefit from medium-scale parallelization, demonstrating relatively linear



scaling up to around 10^2 cores, providing roughly 10^{11} FLOPS. In a typical MD calculation, the computation of the force acting on each atom, which can be the most time-consuming step depending on the complexity and range of the force field, requires N_{step} steps being approximately 10 to 10^2 FLOPS. Therefore, for a typical simulation size of $N = 10^5$ atoms with a timestep in the fs range, we find the ratio of simulated time to real-time on the order

$$\frac{\text{FLOPS}}{N \cdot N_{\text{step}} \cdot dt^{-1}} \sim \frac{10^{11}}{10^5 \cdot 10 \cdot 10^{15}} = 10^{-10}.$$

Hence, the simulation can make progress at a rate of 100 ps per second, or roughly 1 μs per day. This serves as a rough estimate for the capabilities of MD simulations. Regarding the “realism” of the simulations, some general weaknesses include the lack of considering quantum effects and a reduction in the number of dissipation channels available.

3.5 LAMMPS

For the implementation of our MD simulations we use LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator) [20]. LAMMPS provides a numerical framework for setting up the system and integrating the equations of motion with easy access to parallelization of the simulations. This essentially allows us to focus on the higher-level features of the numerical procedure rather than writing the code from scratch.

Chapter 4

Machine Learning

We will use machine learning to predict the friction resulting from the straining and loading of a given Kirigami sheet. To this end, we will generate data through MD simulations that will serve as the ground truth for training a machine learning model. The advantage of using machine learning for this purpose is that it can significantly speed up the exploration of new configurations compared to that using full MD simulations. However, there is no guarantee that the machine learning model can accurately capture the physical mechanisms governing our system. Hence, a key objective is to assess the viability of this approach in the study of Kirigami friction, which we will pursue using traditional machine-learning methods. In this chapter, we introduce the key concept behind machine learning and some of the concepts and techniques relevant to our implementation. For the numerical implementation, we will use the machine learning framework PyTorch [23].

4.1 Neural network

The neural network, or more precisely the *feed-forward dense neural network*, is one of the original concepts in machine learning arising from the attempt of mimicking the way neurons work in the brain [111, 112]. The neural network can be considered in terms of three major parts: The input layer, the so-called *hidden layers* and the output layer as shown in Fig. 4.1. The input is described as a vector $\mathbf{x} = x_0, x_1, \dots, x_{n_x}$ where each input x_i is usually denoted as a *feature*. The input features are densely connected to each of the *nodes* in the first hidden layer as indicated by the straight lines in Fig. 4.1. Each line represents a weighted connection that can be adjusted to configure the importance of that feature. Similar dense connections are present throughout the hidden layers to the final output layer. For a given node $a_j^{[l]}$ in layer l the input from all nodes in the previous layer $l - 1$ are processed as

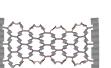
$$a_j^{[l]} = f \left(\sum_i w_{ij}^{[l]} a_i^{[l-1]} + b_j^{[l]} \right),$$

where $w_{ij}^{[l]}$ is the weight connection node $a_i^{[l-1]}$ of the previous layer to the node $a_j^{[l]}$ in the current layer. Note that having the weight belong to layer l as opposed to $l - 1$ is simply a notation choice. $b_j^{[l]}$ denotes a bias and $f(\cdot)$ is the so-called *activation function*. The activation function provides a non-linear mapping of the input to each node. Without this, the network will only be capable of approximate linear functions [111]. Two common activation functions are the *sigmoid*, mapping the input to the range $(0, 1)$, and the *ReLU* which cuts off negative contributions

$$\text{Sigmoid: } f(z) = \frac{1}{1 + e^{-z}}, \quad \text{ReLU: } f(z) = \begin{cases} z & z > 0 \\ 0 & z \leq 0 \end{cases}.$$

Often the same activation function is used throughout a network, except for the output layer where the activation function is usually omitted or the sigmoid is used for classification tasks. The whole process of sending data through the model is called *forward propagation* and constitutes the mechanism for mapping an input \mathbf{x} to the model output $\hat{\mathbf{y}}$. In order to get useful predictions we must *train* the model which involves tuning the model parameters, i.e. the weights and biases.

The model training relies on two core concepts: *backpropagation* and *gradient descent* optimization. First, we define the error associated with a model prediction, otherwise known as the *loss*, through the *loss function*



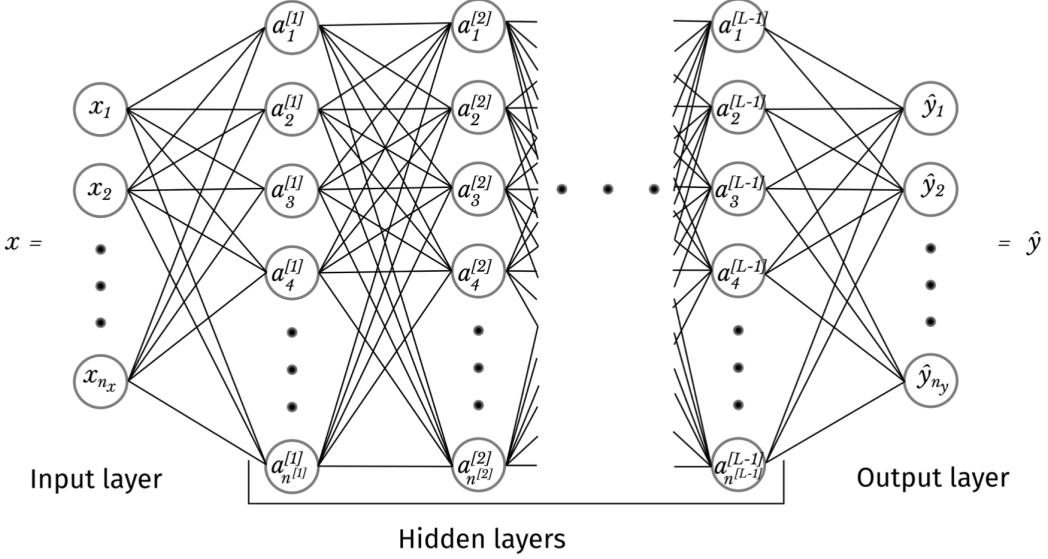


Figure 4.1: Illustration of a general feed-forward dense neural network with n_x input features and n_y outputs. Reproduced from [113].

$L(\hat{\mathbf{y}}, \mathbf{y})$ that evaluates the model output $\hat{\mathbf{y}}$ against the ground truth \mathbf{y} . For a continuous scalar output, we might simply use the mean squared error (MSE)

$$L_{\text{MSE}} = \frac{1}{n_y} \sum_{i=1}^{n_y} (y_i - \hat{y}_i)^2. \quad (4.1)$$

For a binary classification problem, meaning that the output is either True or False (1 or 0), a common choice is binary cross entropy (BCE)

$$L_{\text{BCE}} = - \sum_{i=1}^{n_y} \left[y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \right] = \sum_{i=1}^{n_y} \begin{cases} -\log(\hat{y}_i), & y_i = 1 \\ -\log(1 - \hat{y}_i), & y_i = 0. \end{cases} \quad (4.2)$$

Without going into details with the derivation we can convince ourselves that the error is minimized for the correct prediction and maximized for the worst prediction. When the ground truth is $y_i = 1$ we get the negative loss contribution $-\log(\hat{y}_i)$ where a correct prediction $\hat{y}_i \rightarrow 1$ yields $L_i \rightarrow 0$. For a wrong prediction $\hat{y}_i \rightarrow 0$ the loss contribution will diverge $L_i \rightarrow \infty$. Similar applies to the case of $y_i = 0$ with opposite directions.

Given a loss function, we can calculate the loss gradient $\nabla_{\theta} L$ with respect to each of the model parameters θ , being the weights and biases in the model. This is called *backpropagation* since we follow the propagation of the errors as we go back through the model layers. We calculate the gradients using the derivative chain rule. These gradients express how each parameter is connected to the loss and the overall idea is then to “nudge” each parameter in the right direction to reduce the loss. We usually denote a full cycle of forward propagation, backpropagation and an update of all model parameters as one *epoch*. We calculate the updated parameter θ_t for epoch t using the *gradient descent* method

$$\theta_t = \theta_{t-1} - \eta \nabla_{\theta} L(\theta_t). \quad (4.3)$$

Gradient descent is analog to taking a step in parameter space in the direction that yields the biggest decrease in the loss. If we imagine a simplified case with only two parameters θ_1 and θ_2 we can think of these as longitude and latitude coordinates on a map and the loss being the terrain height. The negative gradient $-\nabla_{\theta} L(\theta_t)$ represents the direction of the steepest loss decline perpendicular to the contour lines shaped by the loss function terrain as shown in Fig. 4.2. Notice, however, that state-of-the-art models in general contain on the order of 10^6 – 10^9 parameters [114] which poses some challenges for the visualization. The length of each step is proportional to the gradient norm $\|\nabla_{\theta} L(\theta_t)\|$ and the learning rate η . There are three main flavors to the gradient descent: Batch, stochastic and mini-batch gradient descent. In *batch gradient descent* we simply calculate the gradient based on the entire dataset by averaging the contribution from each

data point before updating the parameters. This gives the most robust estimate of the gradient and thus the most direct path through parameter space in terms of minimizing the loss function as indicated in Fig. 4.2a. However, for big datasets, this calculation can be computationally heavy as it must carry the entire dataset in memory at once. A solution to this issue is provided by *stochastic gradient descent* (SGD) which considers only one data point at a time. Each data point is chosen randomly with replacement and the parameters are updated based on the corresponding gradient. This leads to more frequent updates of the parameters and a more “noisy” path through parameter space as shown in Fig. 4.2b. Under some circumstances, this might compromise the precision. However, the presence of noise can increase the likelihood of avoiding local minima in parameter space. The *mini-batch gradient descent* serves as a middle ground between the above-mentioned methods by dividing the full dataset into a subset of mini-batches. Each parameter update is then based on the gradient within a mini-batch. By choosing a suitable batch size we get the robustness of the (full) batch gradient descent and the computational efficiency and resistance to local minima of the SGD method. We will use the mini-batch method for our implementation.

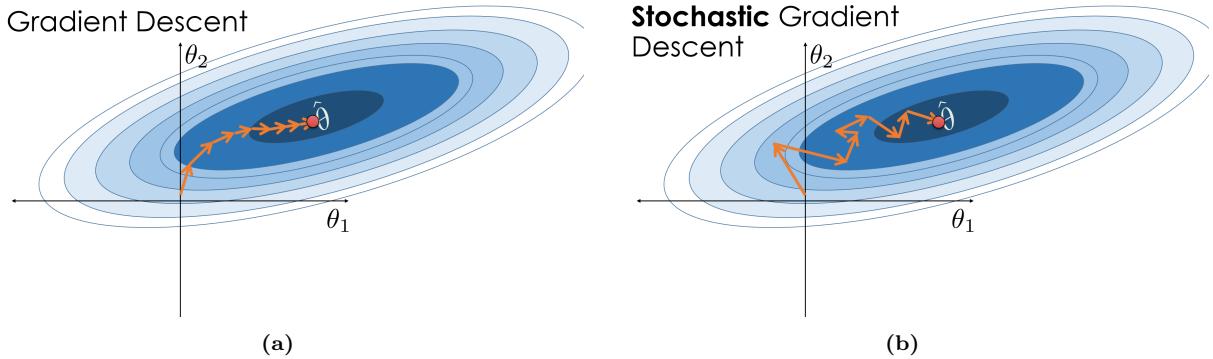


Figure 4.2: Qualitative illustration of the gradient descent method for a simplified problem with only two parameters θ_1 and θ_2 . The blue shade and lines indicate the contour map of the loss function with the darker shade denoting a lower loss. (a) The batch gradient descent method resulting in a relatively straight path toward the optimal parameters. (b) The stochastic gradient descent method resulting in a more noisy path toward the optimal parameters. Reproduced from [115].

4.1.1 Optimizers

The name *optimizers* generally refers to a variety of gradient descent methods. In our study, we will use the ADAM (adaptive moment estimation) optimizer [116]. ADAM combines several “tricks in the book” which we will introduce in the following.

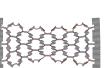
One considerable extension of the gradient descent scheme is the introduction of a momentum term m_t such that we get

$$\theta_t = \theta_{t-1} - m_t, \quad m_t = \alpha m_{t-1} + \eta \nabla_\theta L(\theta_t), \quad (4.4)$$

with $m_0 = 0$. If we introduce the shorthand $g_t = \nabla_\theta L(\theta_t)$ we find

$$\begin{aligned} m_1 &= \alpha m_0 + \eta g_1 = \eta g_1 \\ m_2 &= \alpha m_1 + \eta g_2 = \alpha^1 \eta g_1 + \eta g_2 \\ m_3 &= \alpha m_2 + \eta g_3 = \alpha^2 \eta g_1 + \alpha \eta g_2 + \eta g_3 \\ &\vdots \\ m_t &= \eta \left(\sum_{k=1}^t \alpha^{t-k} g_k \right). \end{aligned} \quad (4.5)$$

Hence m_t is a weighted average of the gradients with an exponentially decreasing weight. This act as a memory of the previous gradients and aid to pass local minima and to some degree plateaus in the parameter space. It also provides a general steadiness to the gradient descent which counteracts the transition from batch to mini-batch gradient descent. A variation of momentum can be achieved with the introduction of



the exponential moving average (EMA) which builds on the recursion

$$\begin{aligned} \text{EMA}(g_1) &= \alpha \overbrace{\text{EMA}(g_0)}^{\equiv 0} + (1 - \alpha)g_1 \\ \text{EMA}(g_2) &= \alpha \text{EMA}(g_1) + (1 - \alpha)g_2 \\ &\vdots \\ \text{EMA}(g_t) &= \alpha \text{EMA}(g_{t-1}) + (1 - \alpha)g_t = \sum_{k=0}^t \alpha^{t-k} (1 - \alpha) g_t, \end{aligned}$$

which is similar to that of momentum Eq. (4.5), but with the explicit weighting by $(1 - \alpha)$. The second moment of the exponential moving average is utilized in the root mean square propagation method (RMSProp) which is motivated by the issue of passing long loss plateaus in the parameter space. Since the size of the updates are otherwise proportional to the norm of the gradient

$$\theta_{t+1} = \theta_t - \eta g_t \implies \|\theta_{t+1} - \theta_t\| = \eta \|g_t\|,$$

we might get the idea of normalizing the gradient step by the norm $\|g_t\|$. However, this does not immediately solve the problem of long plateaus as we need to consider multiple past gradients. Hence, this can be done with the use of the EMA. When reentering a steep region again we need to “quickly” downscale the gradient steps which can be achieved more efficiently by using the squared norm $\|g_t\|^2$ for the EMA which makes it more sensitive to outliers. From this motivation, the RMSProp update scheme is given

$$\theta_t = \theta_{t-1} - \eta \frac{g_t}{\sqrt{\text{EMA}(\|g_t\|^2)} + \epsilon}, \quad (4.6)$$

where ϵ is simply a small number to avoid division by zero issues and thus ensure numerical stability.

ADAM merges the idea of a first-order EMA for the momentum m_t , and a second-order EMA v_t for gradient normalization similar to the root mean square propagation technique in Eq. (4.6)

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1 - \beta_1) g_t, \\ v_t &= \beta_2 v_{t-1} + (1 - \beta_2) g_t^2. \end{aligned}$$

Since m_t and v_t are initially set to zero ADAM introduces the scaling terms $(1 - \beta_1^t)$ and $(1 - \beta_2^t)$ to correct for a bias towards zero. The ADAM scheme is given [116]

$$\theta_{t+1} = \theta_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}, \quad \hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t}. \quad (4.7)$$

4.1.2 Weight decay

By adding a so-called *regularization* to the loss function we can penalize high magnitudes of the model parameters, usually intended for the model weights. This is motivated by the idea of preventing overfitting during training, which we will address in more detail in Sec. 4.3. The most common way to regularize the loss function is by the use of L2 regularization, adding the squared l^2 norm $\|\theta\|_2^2$, where $\|\theta\|_2 = \sqrt{\theta_1^2 + \theta_2^2 + \dots}$, to the model. The loss and gradient then become

$$L_{l^2}(\theta) = L(\theta) + \frac{1}{2} \lambda \|\theta\|_2^2 \quad (4.8)$$

$$\nabla_\theta L_{l^2}(\theta) = \nabla_\theta L(\theta) + \lambda \theta, \quad (4.9)$$

where $\lambda \in [0, 1]$ is the weight decay parameter. The name *weight decay* relates to the fact that some practitioners only apply this penalty to the weights in the model, but we will include the biases as well (standard in PyTorch). Following the original gradient descent scheme Eq. (4.9) we get

$$\theta_{t+1} = \theta_t - \eta g_t - \eta \lambda \theta_t = \theta_t \underbrace{(1 - \eta \lambda)}_{\text{weight decay}} - \eta g_t. \quad (4.10)$$

We notice that choosing a large weight decay ($\lambda \rightarrow 1$) will downscale the model parameters while choosing a low weight decay ($\lambda \rightarrow 0$) yields the original gradient descent scheme. Note that we will use the weight decay principle in combination with ADAM. In Eq. (4.10) we have simply used the original gradient descent scheme Eq. (4.3) since this makes it easier to demonstrate the consequences of introducing the L2 regularization into the loss function Eq. (4.8).

4.1.3 Parameter distributions

In order to get optimal training conditions it has been found that the initial state of the weights and biases are important [117]. First of all, we must initialize the weights by sampling from some distribution. If the weights are set to equal values the gradient across a layer would be the same. This results in a complexity reduction as the model can only encode the same values across the layer. Further, we want to consider the gradient flow during training. Especially for deep networks, networks with many layers, we must pay attention to the problem of *vanishing* or *exploding* gradients. If we for instance consider the sigmoid activation function and its derivative

$$f(z) = \frac{1}{1 + e^{-z}}, \quad f'(z) = \frac{df(z)}{dz} = \frac{e^{-z}}{(1 + e^{-z})^2} = \frac{e^z}{(1 + e^z)^2},$$

we notice that for large and small input values z we get $f(z \rightarrow \pm\infty) \rightarrow 0$. However, even a small finite gradient can vanish throughout a deep network as the calculation of the gradient involves the chain rule. This gives rise to a gradient that potentially gets smaller and smaller for each layer it passes in the backpropagation. A similar problem can be found with the ReLU activation function which contributes toward a gradient of zero for inputs $z < 0$. This can be mitigated by the so-called leaky ReLU which maps the $z < 0$ to a small negative slope $a < 0$ as $f(z) = az$. On the other hand, we have exploding gradients, which are simply a result of the chain rule gradient calculation. For a sufficiently deep network, the gradient can grow exponentially large and sometimes result in a numerical overflow. One approach to mitigate this issue is with the use of gradient clipping, where all gradients above a certain value are manually set to a predefined maximum number. While there exist techniques to accommodate the problem of vanishing or exploding gradients as mentioned above, they both benefit from a properly initialized set of weights. That is, we want the gradients across a given layer to have a zero mean while the variance is similar between layers in the model. This balanced gradient flow is more likely to happen if we initialize the weight by the same criteria [117]. The specific actions to achieve this depends on the model architecture, including the choice of activation functions. For instance, using the ReLU activation functions it was found that the node standard deviation will depend on the number of input nodes from the previous layer $N^{[l-1]}$ as $\sqrt{N^{[l-1]}/2}$ [118]. Thus we can sample the weights from a zero mean normal distribution $N \sim (0, 2/N^{[l-1]})$ with a standard deviation $\sqrt{2/N^{[l-1]}}$ to ensure a balanced weight initialization. This particular case (using the ReLU activation function) is part of the Kaiming initialization scheme [118] which is standard in Pytorch. The bias is initialized from a similar consideration.

Batch normalization is another technique that can help reduce the issue of poor gradient flow. Furthermore, it can benefit by speeding up convergence and making the training process more stable [119]. In general, model parameters are modified throughout training meaning that the range of values coming from a previous layer will shift (internal covariate shift), even though the same training data is fed through the network repeatedly. By scaling the inputs for a given layer, for each mini-batch, we can mitigate this problem and make for a more standardized input range. For layer l we calculate the mean $\mu^{[l]}$ and variance $(\sigma^{[l]})^2$ across the layer with nodes $x_1^{[l]}, x_2^{[l]}, \dots, x_m^{[l]}$ for each mini-batch of size m as

$$\mu^{[l]} = \frac{1}{m} \sum_i^m x_i^{[l]}, \quad (\sigma^{[l]})^2 = \frac{1}{m} \sum_i^m (x_i^{[l]} - \mu^{[l]})^2.$$

We then perform a normal scaling of the inputs within the batch

$$\hat{x}_i^{[l]} = \frac{x_i^{[l]} - \mu^{[l]}}{\sqrt{(\sigma^{[l]})^2 + \epsilon}},$$

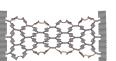
where ϵ is a small number to ensure numerical stability (similar to what we used for RMSProp gradient descent). In the final step, the input values are rescaled as

$$\tilde{x}_i^{[l]} = \gamma^{[l]} \hat{x}_i^{[l]} + \beta^{[l]}$$

with trainable parameters γ and β [119].

4.1.4 Learning rate decay strategies

Until now we have assumed a constant learning rate, but many training schemes use a changing learning rate beyond the adaptiveness included in the optimizers covered so far. Under some circumstances, it can be beneficial to start with a higher learning rate to speed up the initial part of training and then lower the



learning rate for the final part [120]. One straightforward strategy is a step-wise learning rate decay where the learning rate is reduced by a factor $\gamma \in (0, 1)$ every K steps. A more smooth decrease can be achieved with a polynomial decay $\eta_t = \eta_0/t^\alpha$ for $\alpha > 0$. More advanced approaches use multiple cycles of increasing and decreasing cycles. We will mainly concern ourselves with a one-cycle policy for which we start at an intermediate value, increase toward a maximum bound and then decrease toward a final lower learning rate bound. We do this by following a cosine function that is shifted and scaled to increase towards the maximum bound for the first 30% of the training length and decrease toward the lower learning rate bound for the remaining epochs.

4.2 Convolutional Neural Network

Convolutional Neural Networks (CNNs) build upon many of the same concepts as introduced with the feed-forward neural network in Sec. 4.1. The difference lies in its specialization for a spatially correlated input, such as pixels in an image. In a dense neural network, every node is connected to each of the nodes from the previous layers which is not ideal for image recognition. For instance, if we want the model to recognize images of animals the dense network will be very sensitive to where that animal is placed within the frame. The CNN is motivated by the idea of capturing spatial relations in the input, but without being sensitive to the relative placement within the input, i.e. being translational invariant. This is achieved by having a so-called *kernel* or *filter* which slides over the images⁴ as it processes the input. The overall flow of data for a typical convolutional network including a final fully connected neural network is illustrated in Fig. 4.3.

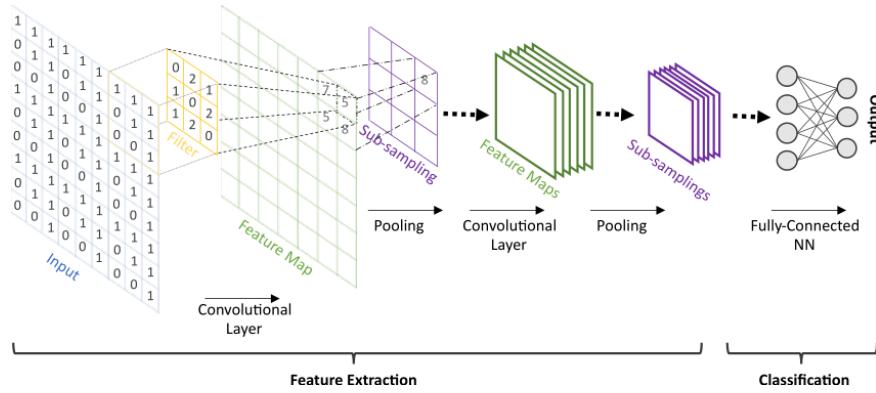


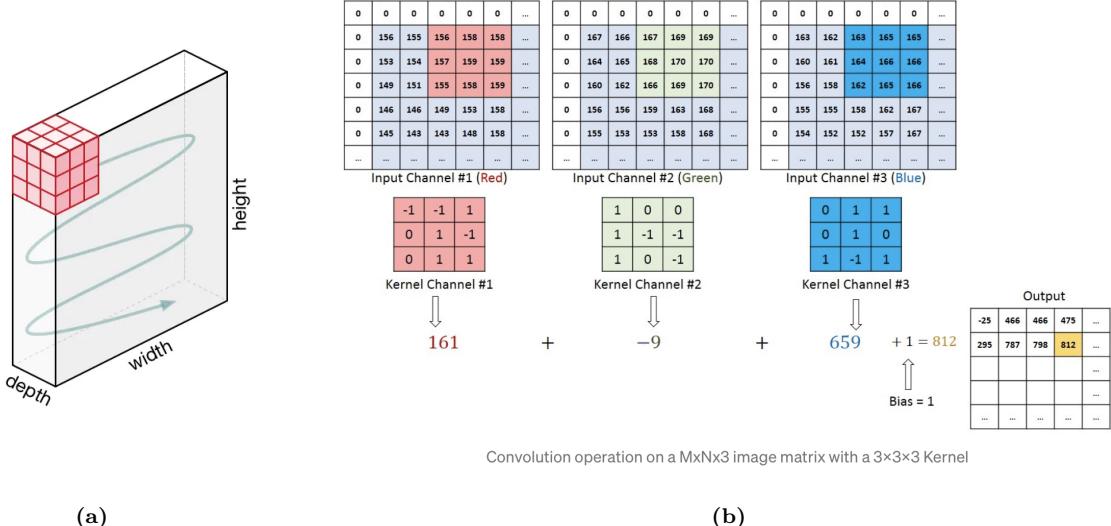
Figure 4.3: Representation of a Convolutional Neural Network (CNN). The CNN performs automatic spatial feature extraction from images by successively applying feature filters that create feature maps (Convolutional layer) and compressing these maps (Pooling layer). Based on the final feature maps, a fully-connected neural network does a prediction, which can be a classification or regression. Figure and caption reproduced from [121].

A convolutional layer contains multiple kernels, each consisting of a set of trainable weights and a bias. Each kernel will produce a separate output channel to the resulting *feature map* layer. The kernel has a 2D spatial size, specific to the model architecture, and a depth that matches the number of input channels to the layer. For instance, a typical RGB image will have three channels, while the number of channels usually increases for each layer in the model. The kernel lines up with the image and calculates the feature map output as a dot product between the weights in the kernel and the aligning subset of the input. This is done for each input channel and summed up with the addition of a bias as illustrated in Fig. 4.4. The kernel then slides over by a step size given by the *stride* parameter and repeats the calculation. Choosing a stride of 2 or higher results in a reduction of the output spatial size. If we want to preserve the spatial size we must keep a stride of one and additionally apply *padding* to the input images, such that we can achieve one kernel position for each input “pixel”. The spatial size of the feature map is given as

$$N_d^{[l]} = \left\lfloor \frac{N_d^{[l-1]} - F_d + 2P}{S} + 1 \right\rfloor, \quad (4.11)$$

for padding P , stride S , spatial size of the kernel filter F_d , spatial size of the input $N_d^{[l-1]}$, for dimension $d = \{x, y\}$ and layer l .

⁴Note, that we will be using the word “image” as a reference for a spatially dependent input, but in reality, it does not have to be an actual image in the classical sense.



(a)

(b)

Figure 4.4: Illustration of the convolution procedure for a 3×3 kernel with a depth of 3 matching an RGB input with 3 channels (a) The kernel goes through the various positions aligning with the input as sketched with the arrow. The precise number of positions along this path depends on the stride parameter and the choice of padding. (b) A specific example of the calculations involved in the convolution process. Here a rim of zero padding is included. For each kernel position, a dot product is calculated between the aligning input and the kernel weights for each corresponding channel. This results in an output value for each channel which is then summed up and added with a bias to constitute the final output in the feature map. Reproduced from [122].

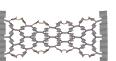
The *down-sampling* is often done through a pooling layer. A pooling layer is reminiscent of a kernel, but instead of calculating the output as a dot product, it calculates the mean (mean pooling) or the max value (max pooling) of the values within its scope. For instance, by using a max pooling of size 2×2 and stride 2 we essentially half the dimensions of the image as dictated by Eq. (4.11). CNNs will often use repeating series of convolution (applying a kernel), pooling and then an activation function. Most architectures aim to down-sample the spatial input while increasing the number of channels throughout the model layers. This results in a smaller set of features extracted from the input which then can then be fed into a dense network, or *fully connected* connected neural network, as also illustrated in Fig. 4.3. The convolution part aims to handle the transition from a spatial input to some internal features. For a model which recognizes animals, we would perhaps think of features such as the number of legs, size, color and so on. In practice, however, the network will not give readily interpreted features for the processing in the fully connected layer, but the concept is still the same.

For a CNN, we often consider the *receptive field*. The receptive field relates to the spatial size of the input that affects a given node in the feature map of a given layer in the model. This term is often used in reference to the output nodes. Fig. 4.5 illustrates the receptive field for a 1D representation of a CNN with repetitive use of a kernel of width 2 and stride 1. Going from the output and backward, we see that the output layers are connected to two nodes in the previous layer. Each of these nodes is connected to two nodes in the layer before that, however with one of them overlapping due to the stride of 1. By back-tracking all the way to the input layer we see that this corresponds to a receptive field of $D = 5$. This means that a single output node is only affected by the 5 input nodes within its receptive field. By increasing the filter size and the stride the receptive field will grow a lot faster than shown in this example. If we assume a constant filter size throughout the network F , a stride S_l from layer $l - 1$ to l we get that the receptive field D_l with respect to a certain layer in a given spatial dimension is

$$D_l = D_{l-1} + \left[(F_l - 1) \cdot \prod_{i=1}^{l-1} S_i \right], \quad (4.12)$$

with $D_0 = 1$ and $l = 0$ as the input layer. Note that by convention, the product of zero elements is 1, such that for the first layer, the product is 1. Eq. (4.12) apply for both spatial dimensions individually.

The receptive field is important for understanding the connectivity in the model since the output will be completely independent of the inputs and feature maps outside the receptive field. Furthermore, we differentiate between the *theoretical* receptive field and the *effective* receptive field. The effective receptive



field will have a Gaussian distribution within the theoretical receptive field because the nodes in the center of the receptive field will have more connections leading to the output, as seen in Fig. 4.5. Thus, in practice, the effective receptive field will be smaller than the theoretical. Implementations like dilated convolutions, which make the filter expand in circumference and skip positions within the filter, can be used to further increase the effective field.

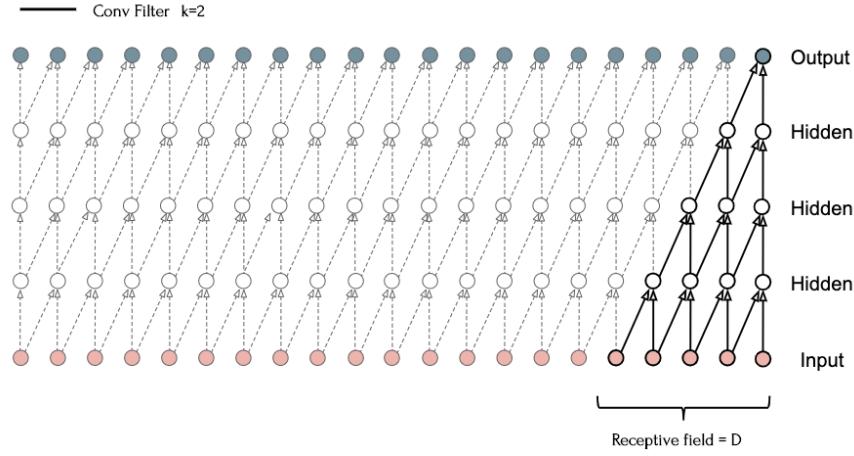


Figure 4.5: An illustration of the receptive field D with respect to an output node in a 1D convolutional network. This example uses a width of 2 for the kernel and a stride of 1. Reproduced from [123].

4.2.1 Training, validation and test data

So far, we have simply considered the concept of *training data* as a means to update the model parameters. Yet, we want to evaluate the model performance as it improves. The problem arises immediately from the fact that a complex model can fit about any function. More precisely, it has been proven that a deep convolutional neural network is universal (it follows the universal approximation theorem), meaning that it can approximate any continuous function to an arbitrary accuracy when the depth of the network is large enough [124]. Thus for a complex model, it is just a matter of time (epochs) before the model eventually finds a good approximation for the training data. However, we want the model to learn general trends and not to “memorize” all the data points which are known as *overfitting*. While the predictions for the training data can grow arbitrarily good in most cases, the performance on unseen data within the same domain will yield poor performance in the case of overfitting. The common way to address this issue is by putting aside a subset of the data, the so-called *validation* data, which we use to validate the model performance during and after training. By keeping this validation set separate from the training data we can get a more reliable performance estimate for the model. It is crucial to use a random partitioning in order to ensure an equal distribution of data across both sets. To strike a balance between the quality of training and validation, a commonly used partitioning ratio is usually around 20:80 in favor of the training set which we will adapt as well. A third data set often forgotten is the *test* set. While the validation set should be kept unseen from the model training, the test set should be kept unseen from the model developer when choosing the model architecture and hyper-parameters. We define a hyper-parameter as a variable to be set prior to the actual application of the learning algorithm, one that is not selected by the learning algorithm itself [125]. This includes parameters such as learning rate, momentum and weight decay, but not the weights and biases as these are updated by the learning algorithm. When adjusting the hyper-parameters we will use the performance on the validation set as a guiding metric. Hence, our choices can eventually lead to high-level overfitting through the hyper-parameter choices. Hence, we should denote a test set for the final evaluation of our model which has not been considered before the end. Formally, this is the only reliable performance metric for the model. However, it is important that this set also possesses the same data distribution to ensure a reliable performance estimate.

4.3 Overfitting and underfitting

The balance between underfitting and overfitting is essential in model training and it is closely related to the complexity of the model and the selected hyper-parameters. A typical textbook visualization of underfitting and overfitting is given in Fig. 4.6a.

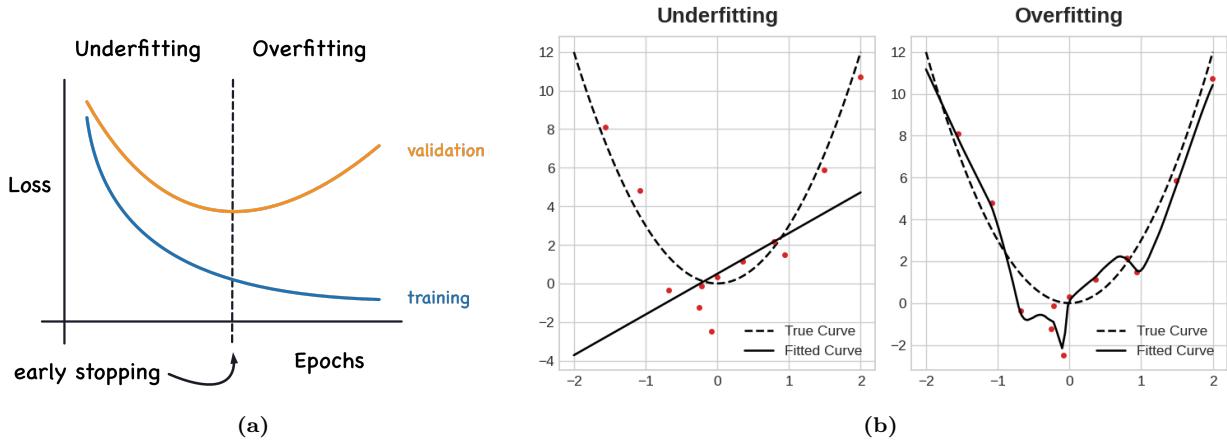


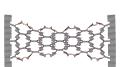
Figure 4.6: Illustrations regarding the concept of underfitting and overfitting. (a) A typical textbook representation of the transition from underfitting to overfitting during prolonged training. Initially, both the training and validation loss decreases for each epoch, but at some point, the validation will begin to increase again marking the beginning of overfitting. This motivates the idea of an early stopping point at the minimum of the validation loss curve. (b) A 2D example of fitting a curve (solid line) to a series of data points (red dots) generated from a second-order polynomial (dotted line) with additional noise. In the underfitting case, we fail to capture the general trend due to a low complexity in our fit, while in the overfitting case, our fit captures the noise in the data as well. Reproduced from [126].

As we begin to train or model both the training and validation loss is decreasing. At a certain point, the model may begin to not only capture general trends but also specific trends in the training data, which may be related to noise. This marks the transition from underfitting to overfitting where the validation loss will increase due to a loss of generalization. However, the training loss will continue its steady decline as the training procedure optimize exclusively for the training data loss. *Early stopping* can be utilized to detect this transition and stop the training early in an attempt to hit the sweet spot between under- and overfitting. We will use a variation of this technique which is to store the best model based on the validation performance history. For this approach, we let the training finish and then store the model corresponding to the best validation score found during the entirety of the training. In principle, we can “get lucky” and find the model settings at a state that is overfitted for the validation set. However, we consider this to be highly unlikely for a reasonably big amount of data and a complex model with many model parameters to tune.

The underfitting and overfitting phenomena can also be thought of as a consequence of model complexity and not just training time. For a certain amount of epochs a simple model will yield underfitting and an overly complex model will yield overfitting. This can be expected to follow a similar qualitative trend as shown in Fig. 4.6a with the substitution of *model complexity* on the x-axis. Fig. 4.6b visualizes the concept of underfitting and overfitting in terms of the complexity regarding the fitting of a second-order polynomial. The figure shows how a simple linear function will make a crude approximation for the true curve, while an overly complex model will pick up the noise in the data and miss the general trend as well. However, the problem is that we do not know the true curve in practice. If we did, we would not need machine learning to approximate it in the first place. Without having additional insight into the governing source of the data the overfitting case from Fig. 4.6b seems to produce the most confident fit for all we know. Thus, it is important to utilize the validation loss in order to mitigate this problem.

4.4 Hypertuning

The training of a machine learning model revolves around tuning the model parameters such as weights and biases. However, as mentioned already, a handful of *hyper-parameters* remains for us to decide. First of all, we need to choose an architecture for the model. This includes high-level considerations such as whether to use a neural network or a convolutional network, but also lower-level considerations, such as the depth and the width of the model, i.e. the number of layers and the number of nodes/channels in a layer. In addition, we have to choose the loss function and the optimizer which come with hyper-parameters such as learning rate, momentum and weight decay. This extensive list of choices makes the designing of a functional machine learning procedure more complicated than simply hitting “run” for the learning algorithm. As N. Smith [120] puts it “Setting the hyper-parameters remains a black art that requires years of experience to acquire”. In the following, we will review a general approach for choosing the learning rate, momentum and weight decay



hyper-parameters based on the findings of [120]. The traditional approach is to perform a *grid search*, trying out different combinations of hyper-parameters in different training sessions, but this might rather quickly become computationally expensive and ineffective. In addition, hyper-parameters will depend on the training data, the model architecture and not at least each other. This makes it difficult to narrow down the choice one by one. N. Smith points to the fact that the validation loss can be examined early on for clues of either underfitting or overfitting.

The learning rate is often regarded as the most important hyper-parameter to tune [125]. Typical values are in the range $[10^{-6}, 1]$. Instead of simply running a grid search, we can perform a so-called *learning rate range test*. One then specifies the minimum and maximum learning rate boundaries for the range test and a learning rate step size. A range of 10^{-7} to 10 will most likely be appropriate, but the test will reveal this immediately. The idea is to vary the learning rate throughout the given range in small steps during a short pre-training. We will increase the learning rate for each iteration, i.e. each parameter update following a mini-batch, and thus we can run this test for a few epochs, or even a single one, depending on the number of mini-batches. For small learning rates, the model will converge slowly. As the learning rate approaches an appropriate value, the convergence speed will increase, which can be observed as a decrease in the validation loss. Eventually, the convergence will stop and the validation loss will pass a minimum for which it will begin to diverge. This general behavior can be understood for the simplified 1D example of finding the minimum of a second-order polynomial as shown in Fig. 4.7. For small learning rates, the gradient descent update will provide a small step in the parameter space toward the minimum. The gradient descent update will provide a small step in the parameter space toward the minimum. However, if the learning rate is too small this will yield a slow convergence. On the other hand, if the learning rate becomes too large, we will effectively step past the minimum in one step. Each following step will overshoot the minimum further since the step length is proportional to the gradient of the loss, which leads to a diverging trend. From the learning rate range test, we can use the point of divergence as an upper bound for the learning rates when considering a cyclic learning rate scheme. For a constant learning rate scheme we can use the point of steepest decline of the validation loss as an estimate for the best learning rate choice [120].

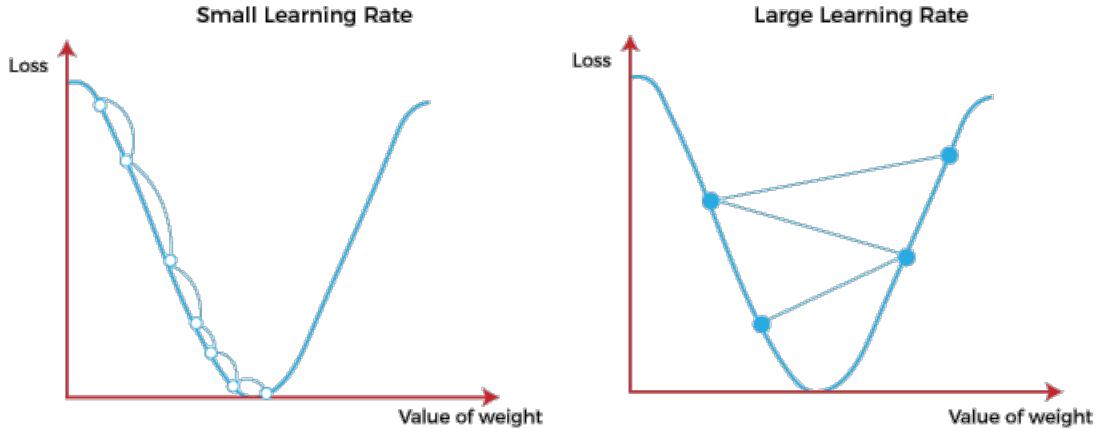


Figure 4.7: Qualitative illustration of the effect of learning rate choice for gradient descent on a simplified 1D problem. The left frame shows the gradient descent steps using a small learning rate where several steps are needed to approach the minimum of the loss. The right frame shows the corresponding steps with the choice of a too large learning rate which yields a diverging behavior. Reproduced from [127].

Next, we consider the choice of momentum. From the gradient descent scheme with momentum Eq. (4.4) we see that the momentum parameter α and the learning rate η have a similar effect on the parameter update

$$\theta_t = \theta_{t-1} - \eta g_t - \alpha m_{t-1},$$

since m_t is a moving average of the gradient g_t as well. Like the learning rate, we want to set the momentum value as high as possible without causing instabilities in the training. However, since the two are related to each other, they must also be considered together. N. Smith [120] reports that a momentum range test is not a useful approach to determine the optimal momentum value. Instead, he suggests doing a few short runs with different values of momentum, such as 0.99, 0.97, 0.95, and 0.9, to determine a suitable choice. By including momentum in the learning rate test we can balance the learning rate accordingly. Moreover, for a cyclic learning rate scheme he reports that a cycling momentum scheme, reversed with respect to the learning rate, is beneficial. When the learning rate increase toward the upper bound the learning rate should decrease toward a lower bound and vice versa. Choosing a lower momentum bound of 0.80–0.85 is reported to give similar stable results [120].

Finally, we address weight decay. N. Smith [120] reports that weight decay is different from learning rate and momentum by the fact that weight decay is better chosen as a constant value as opposed to a cyclic scheme. However, the weight decay is dependent on the model complexity, learning rate and momentum choice. Hence, this can often be chosen after setting those. We can estimate a suitable choice by doing a rough grid search for values such as 0, 10^{-6} , 10^{-5} and 10^{-4} for complex architectures and 10^{-4} , 10^{-3} and 10^{-2} for more shallow architectures. Choosing the weight decay on the scale of exponents will often provide good enough precision in practice.

4.5 Prediction explanation

On a final note, we present a simple method for providing some insight into the prediction from a convolutional neural network. The high complexity of deep learning models limits our ability to understand the decision-making going into the model predictions beyond the input data. This is known as the *black box* problem. A lot of effort is currently being developed for making more transparent models, like decision trees with interpretable rules, and numerical tools for unpacking the inner workings of the model. We will consider a gradient-based method called *Grad-CAM* [128] which aims to highlight some of the important features of a convolutional network model prediction. The method utilizes the gradients for a certain feature map with respect to the loss.

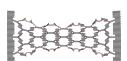
To begin, we forward propagate the input through the model. Next, we decide on a feature map of interest and calculate the gradients with respect to the loss of a certain target output. In classification tasks, one would often choose the predicted class with the highest score. By doing so, the gradients can be utilized to determine which parts of the feature map are more important for the prediction. We apply a ReLU activation layer to keep only the positive contributions. Since the convolutional layers preserve spatial information we can rescale the feature map gradients to make an input-sized heatmap allowing for an overlaid visualization on top of the input image. By using this method, we can obtain a visual indication of the specific regions in the image that plays a key role in the prediction. We can apply this technique to multiple layers of the model and even merge the outcomes as well for an analysis of the contribution from multiple feature maps.

Fig. 4.8 show an exemplary use, where the Grad-CAM analysis reveals the difference between a biased and unbiased model for the task of predicting professions. The biased model appears to be prioritizing the person's facial features over the objective cues provided by relevant equipment and work-related clothing.

4.6 Accelerated search using genetic algorithm

For the scope of finding new Kirigami designs which exhibit certain frictional properties, we are interested in utilizing a trained machine-learning model for further exploration. This reverses the design process as one has to find the right input to achieve a certain output. A possible strategy is to explore a range of inputs and use the model predictions as a guiding metric. One approach to this strategy is the genetic algorithm (GA) which is inspired by biological evolution mimicking the Darwin theory of the survival of the fittest [129]. GA is a population-based algorithm for which the basic elements are chromosome representation, fitness selection and biological-inspired operators. The chromosomes represent the genes for each individual in the population and typically take the form of a binary string. Each position within the chromosome is called a *locus* and has two possible values (0 or 1). A fitness function is defined to assign a score for all chromosomes based on some optimization objective. This plays a role for the biologically inspired operators for which the main ones are selection, mutation and crossover. Selection is the process of selecting chromosomes based on their fitness score for further processing. In mutation, some of the loci within a chromosome are changed and in crossover, chromosomes are merged to create offspring. GA has been implemented in many areas such as the traveling salesman problem [130], function optimization [131], adaptive agents in stock markets [132] and airport scheduling [133]. Wang et al. [134] note that a general drawback is a need for expertise when choosing parameters that match specific applications. They propose an accelerated genetic algorithm based on a Markov chain transition probability matrix to perform a guided search that reduces the number of parameter choices one has to make. The following introduction of this method is based on their work [134].

We define the binary population matrix $A_{ij}(t)$ at generation t , consisting of N rows denoting chromosomes $i \in \{0, 1, \dots, N\}$ and L columns denoting the loci $j \in \{0, 1, \dots, L\}$. For our application, we let the locus represent an atom in the Kirigami pattern matrix which is flattened to fit the format of the population matrix. We carry forward the binary values with 0 meaning a removed atom and 1 a present atom. By the use of a fitness function $f(t)$, we sort the population matrix row-wise in descending order by fitness score, i.e. $f_i(t) \leq f_k(t)$ for $i \geq k$. In the spirit of Markov chains, we assume that some transition probability exists for



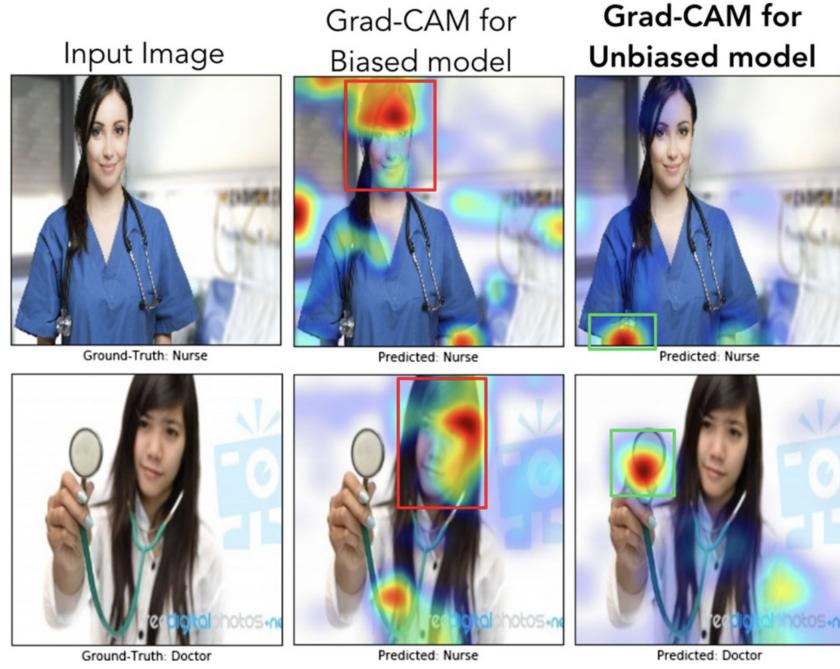


Figure 4.8: An example of the Grad-CAM method in use: In the first row, we can see that even though both models made the right decision, the biased model was looking at the face of the person to decide if the person was a nurse, whereas the unbiased model was looking at the short sleeves to make the decision. For the example image in the second row, the biased model made the wrong prediction (misclassifying a doctor as a nurse) by looking at the face and the hairstyle, whereas the unbiased model made the right prediction looking at the white coat, and the stethoscope. Figure and caption reproduced from [128].

the transition between the current state $A(t)$ and the next state $A(t+1)$. We assume that this transition probability only takes into account the mutation process, and thus we omit operators like crossover. For each generation, the chromosomes are sorted according to the fitness function and the chromosome at the i^{th} fittest place is assigned a ranking score $r_i(t)$ by some monotonic increasing ranking scheme. We take this to be

$$r_i(t) = \begin{cases} (i-1)/N', & i-1 < N', \\ 1, & \text{else} \end{cases},$$

with $N' = N/2$ from [134]. We assign a row mutation probability $a_i(t) = r_i(t)$ meaning that the probability for a mutation will increase towards the lower fitness scores. For the considerations of the mutation probability for the loci within each chromosome, we define the count of 0 and 1 states as $C_0(j)$ and $C_1(j)$ respectively. These are normalized as

$$n_0(j, t) = \frac{C_0(j)}{C_0(j) + C_1(j)}, \quad n_1(j, t) = \frac{C_1(j)}{C_0(j) + C_1(j)}.$$

We can thus describe the state of the j^{th} locus column as the state vector $\mathbf{n}(j, t) = (n_0(j, t), n_1(j, t))$. In order to direct the current population to a preferred state for locus j we consider the highest weight $W_i = 1 - r_i$ among the chromosomes for the case of the locus being 0 or 1 respectively. This corresponds to the targets

$$\begin{aligned} C'_0(j) &= \max\{W_i | A_{ij} = 0; i = 1, \dots, N\} \\ C'_1(j) &= \max\{W_i | A_{ij} = 1; i = 1, \dots, N\}. \end{aligned}$$

These are normalized

$$n_0(j, t+1) = \frac{C'_0(j)}{C'_0(j) + C'_1(j)}, \quad n_1(j, t+1) = \frac{C'_1(j)}{C'_0(j) + C'_1(j)}, \quad (4.13)$$

to produce the target state vector $\mathbf{n}(j, t+1) = (n_0(j, t+1), n_1(j, t+1))$. This will serve as a direction for each locus to evolve in and thus we can formulate the Markov chain as

$$\begin{bmatrix} n_0(j, t+1) \\ n_1(j, t+1) \end{bmatrix} = \begin{bmatrix} P_{00}(j, t) & P_{10}(j, t) \\ P_{01}(j, t) & P_{11}(j, t) \end{bmatrix} \begin{bmatrix} n_0(j, t) \\ n_1(j, t) \end{bmatrix},$$

where the matrix represents the transition matrix. Since the probability must sum to one for the columns in the transition matrix we get

$$P_{01}(j, t) = 1 - P_{00}(j, t), \quad P_{11}(j, t) = 1 - P_{10}(j, t).$$

These conditions allow us to solve for the transition probability $P_{10}(j, t)$ in terms of the single variable $P_{00}(j, t)$

$$P_{10}(j, t) = \frac{n_0(j, t + 1) - P_{00}(j, t)n_0(j, t)}{n_1(j, t)} \quad (4.14)$$

$$P_{01}(j, t) = 1 - P_{00}(j, t) \quad (4.15)$$

$$P_{11}(j, t) = 1 - P_{10}(j, t) \quad (4.16)$$

The remaining part is to define $P_{00}(j, t)$. We adopt the choice from [134] and start from $P_{00}(j, t = 0) = 0.5$ and choose $P_{00}(j, t) = n_0(j, t)$ for the following generations $t > 0$. Thus for a locus $A_{ij}(t)$ we mutate it, changing the binary value, by the probability

$$p_{ij}(t) = \begin{cases} a_i(t)P_{01}(t), & A_{ij}(t) = 0 \\ a_i(t)P_{10}(t), & A_{ij}(t) = 1 \end{cases} \quad (4.17)$$

In summary, each generation update involves the following steps.

1. For generation t calculate the fitness score $f_i(t)$ of each chromosome i and sort the population matrix $A_{ij}(t)$ row-wise with descending score.
2. From the chosen ranking scheme $r_i(t)$ set the chromosome mutation probability to $a_i(t) = r_i(t)$ and the weighting of each row $W_i(t) = 1 - r_i(t)$.
3. Calculate the target states in Eq. (4.13) and the transition probabilities using Eq. (4.14) to (4.16) and $P_{00}(j, t = 0) = 0.5$, $P_{00}(j, t > 0) = n_0(j, t > 0)$.
4. Mutate $(0 \rightarrow 1$ or $1 \rightarrow 0)$ each locus $A_{ij}(t)$ by the probability p_{ij} given by Eq. (4.17).

Notice that this algorithm treats every locus as an independent gene, which means that we do not take the spatial dependencies in the Kirigami pattern into account.



Part II

Simulations



Chapter 5

Creating a graphene kirigami system

The system geometry plays an essential role in the “friction experiment” that we are going to carry out through MD simulations. The purpose of the simulations is to quantify the friction that arises when a stretched Kirigami graphene sheet slides over a substrate. We aim to design the simplest possible system that allows for such a measurement under variations of Kirigami design, strain and load.

For this purpose, two approaches were considered as sketched in Fig. 5.1. One approach is simply to mimic a FFM type experiment where the graphene sheet is resting on a substrate and a moving body scans across the graphene surface as seen in Fig. 5.1b. This setup allows for a variety of tip designs, and we can even substitute the tip for a flat surface making the setup resemble a SFA experiment instead. For this setup, we would attach a pre-stretched sheet to the substrate and require the edges of the sheet to be fixated on the substrate to sustain the strain. Thus, the sheet and substrate would constitute the manufactured object and the moving body would represent the contact to the outside world. In this approach, the potential applications would relate to certain effects being associated with a constant strain value. Another approach is to have the sheet ends fixated on the moving body instead as shown in Fig. 5.1a. This switches the roles of the involved parts as we now view the moving body and the sheet together as the manufactured object, while contact with the substrate represents the outside world. This allows for the introduction of a nanomachine design that converts the loading of the manufactured object into a strain of the sheet. Thus, the possible applications allow for a dynamic effect with changing strain through the loading of the sheet. While both methods serve as novel approaches with prospects of providing valuable insight into a sparsely covered field, we choose the latter option (Fig. 5.1a) due to the increased application possibilities.

We do not attempt to model the nanomachine explicitly, but we will use the conceptual idea of a coupling between load and strain to motivate our study. Hence our system of choice consists of a 2D graphene sheet with locked ends, mimicking the attachment to a moving body, and a 3D silicon bulk substrate.

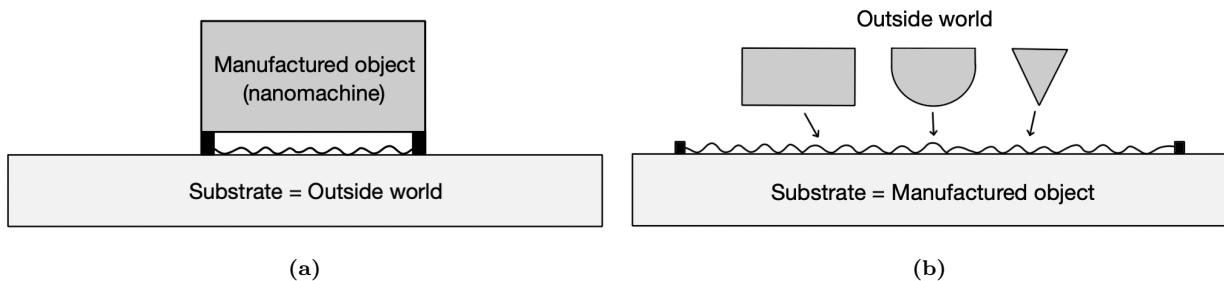


Figure 5.1: Conceptual visualization of two different system setups considered. The wiggly line represents the Kirigami sheet. (a) The chosen setup with a manufactured object connected to the sheet. The contact with the substrate represents the contact with the outside world. (b) An alternative setup where the sheet is fixed in the substrate constituting the manufactured object. The contact with the outside world is then represented through an indentation by objects with various shapes and sizes.



5.1 Region definitions

We subdivide the two main parts of the system, the sheet and the substrate, into specific regions according to their functionality in the MD simulations. For the sheet, we denote a subsection of the ends, with respect to the sliding direction, as so-called *pull blocks*, which is reserved for the application of normal load, stretching, sliding of the sheet, and for applying the thermostat. The remaining *inner sheet* is left for the Kirigami cuts and is simulated as an *NVE* ensemble. This partitioning is motivated by the idea that we want to minimize direct manipulations on the inner sheet, given its presumed critical role in governing friction behavior. The pull blocks are equally split between a thermostat part and a rigid part. It should be noted that the rigidness of the pull blocks is enforced only after a relaxation period to ensure that the crystal structure is fully relaxed. This is further explained in Sec. 5.2. The substrate is equally divided into three parts: The *upper layers* (*NVE*) responsible for the sheet-substrate interaction, the *middle layers* being a thermostat (*NVT*), and the *bottom layers* being frozen, made rigid and fixed, in the initial lattice structure to ensure that the substrate stays in place. Fig. 5.2 shows the system with colors matching the three distinct roles:

1. Red: *NVE* parts which are governing the frictional behavior of interest.
2. Green: Thermostats (*NVT*) surrounding the *NVE* parts in order to modify the temperature without making disturbing changes to the interaction of the sheet and substrate.
3. Blue: Parts that are initially or eventually turned into rigid objects. For the substrate, this refers to an additional fixation as well.

The full sheet is given a size $\sim 130 \times 163 \text{ \AA}$ while the substrate is scaled accordingly to the sheet which is further specified in Sec. 5.3. For an expected strain of 200%, the total system size is roughly 5.7×10^4 atoms. The specific distribution of atoms is shown in Table 5.1 along with the spatial x-y-measures in Table 5.2. An example of a strained sheet is shown in Fig. 5.3.

Table 5.1: Specification of the system size regarding the number of atoms for various system regions. These numbers correspond with the case of no cuts applied to the sheet and a substrate scaled for the expected sheet strain of 200%.

| Region | Total | Sub-region | Subtotal | NVE | NVT | Rigid |
|------------|-------|-------------|----------|-------|-------|-------|
| Full sheet | 8060 | Inner sheet | 6572 | 6572 | 0 | 0 |
| | | Pull blocks | 1488 | 0 | 744 | 744 |
| Substrate | 49068 | Upper | 16356 | 16356 | 0 | 0 |
| | | Middle | 16356 | 0 | 16356 | 0 |
| | | Bottom | 16356 | 0 | 0 | 16356 |
| All | 57128 | | | 22928 | 17100 | 17100 |

Table 5.2: Specification of the spatial size of the system for the x-y-dimensions with a substrate scaled for an expected sheet strain of 200%. The first column denotes the size relative to the full sheet size $x_S \times y_S$, while the second column denotes the corresponding length in \AA .

| Region | Dim | Dim [\AA] | Area [nm^2] |
|-------------|--------------------------------|--|------------------------|
| Full sheet | $x_S \times y_S$ | $130.029 \times 163.219 \text{ \AA}$ | 212.23 |
| Inner sheet | $x_S \times 0.81 y_S$ | $130.029 \times 132.853 \text{ \AA}$ | 172.74 |
| Pull blocks | $2 \times x_S \times 0.09 y_S$ | $2 \times 130.029 \times 15.183 \text{ \AA}$ | 2×19.74 |
| Substrate | $1.16 x_S \times 3.12 y_S$ | $150.709 \times 509.152 \text{ \AA}$ | 767.34 |

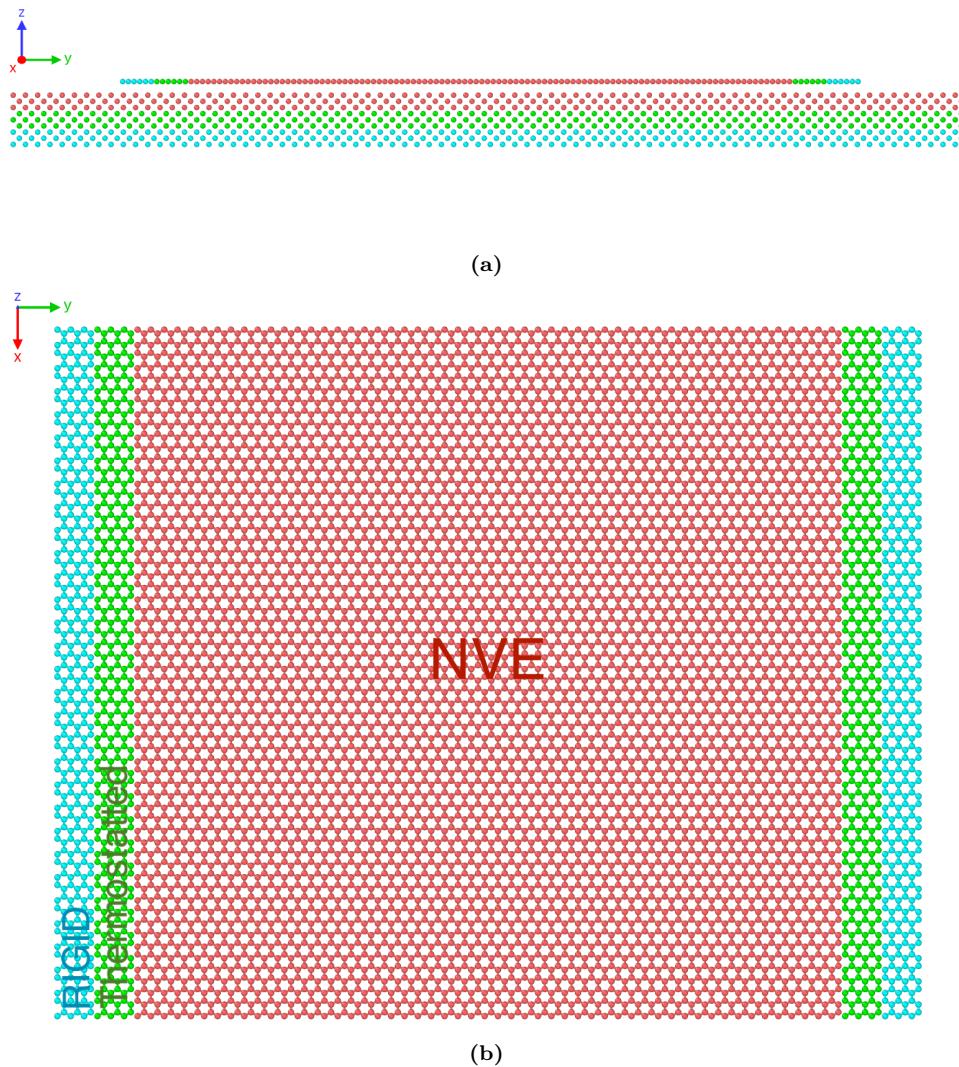


Figure 5.2: System configuration colorized to indicate NVE parts (red), thermostat parts (green) and rigid parts (blue). (a) Side view showing the sheet on top of the substrate. (b) Top view showing only the sheet.

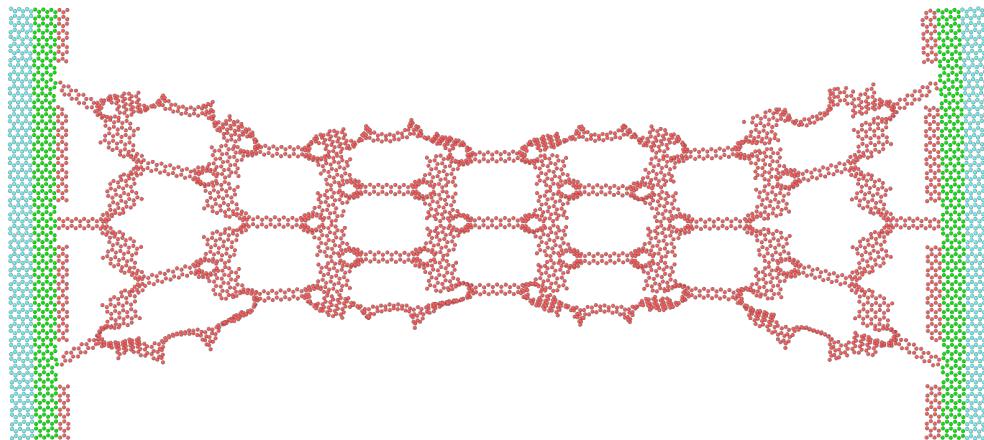


Figure 5.3: Stretched Kirigami sheet against a substrate. The substrate is excluded from the image for better visibility. The atoms are colorized to indicate NVE parts (red), thermostat parts (green) and rigid parts (blue). The pattern used is the Honeycomb (2, 2, 1, 5) (see Chapter 5) at a 100% strain level.

5.2 Numerical procedure

Following the system setup as described previously, the next step involves letting the system relax and reach a stable equilibrium. During this phase, slight modifications to the sheet-substrate distance and lattice spacing take place, both of which are influenced by the thermostat's target temperature. We then stretch the sheet to the desired length, apply normal load and finally slide it along the substrate. The full numerical procedure can be arranged into the following steps. Some steps have been given a default duration denoted in parentheses in units of ps, 10^{-12} seconds.

1. **Relaxation** (15 ps): The sheet and substrate are relaxed for 15 ps after being added in their crystalline form with a separation distance of 3 Å. Given that the equilibrium separation distance will vary with temperature, this value is based on an average estimate suiting our temperature range of interest. In order to avoid any sheet drift we constrain it with the use of three hard spring forces with a spring constant of $10^5 \text{ eV}/\text{\AA}^2 \sim 1.6 \times 10^6 \text{ N/m}$: One spring attaches the sheet center of mass (CM) to its original position, preventing CM drift, while the remaining two springs are attached to the pull blocks CM, to prevent rotation. In principle, fixing only one of the pull blocks would suffice, but we choose to fixate both to maintain symmetry. During the relaxation phase, we consider the pull blocks to be rigid with respect to the z-direction only (perpendicular to the sheet). That is, all the forces in the z-direction are summed up and applied as a uniform external force, while the pull blocks are free to expand and contract in the x-y-plane. This feature is incorporated to allow the pull blocks to readjust the lattice spacing according to the temperature of the system. For the following steps, the pull blocks are made truly rigid with respect to all directions, and the spring forces are terminated.
2. **Stretch**: To stretch the sheet, the two opposing rigid parts of the pull blocks are moved apart at a constant velocity until the desired strain level is achieved. The duration of this phase is determined by the values of the *strain speed* and *strain amount* parameters.
3. **Pause** (5 ps): The sheet is further relaxed for 5 ps to ensure that the sheet is stable and equilibrated after the applied strain deformation.
4. **Normal load** (5 ps): The pull blocks are subjected to a uniformly applied load in the negative z-direction, thereby pushing the sheet perpendicularly into the substrate. Initially a viscous damping force, $F = -\gamma v$ with damping factor γ and velocity v , is added to resist rapid acceleration of the sheet and prevent a hard impact between the sheet and substrate as the sheet-substrate distance decreases. The damping coefficient is set to $\gamma = 8 \times 10^{-4} \text{ nN}/(\text{m/s})$ and terminated after 0.5 ps which was found to be suitable for the extreme load cases of our intended range. The remaining 4.5 ps is devoted to further relaxation in order to reach a sheet-substrate distance equilibrium.
5. **Sliding**: A virtual atom is introduced into the simulation which exclusively interacts with the rigid parts of the pull blocks through a spring force with spring constant K in the x-y-plane. The force in the z-direction is not influenced by the spring force and is instead governed by the equilibrium between the normal load and the normal force response from the sheet-substrate interaction. The virtual atom is immediately given a constant velocity, given by the *sliding speed* parameter. This results in an initial linear increase in sliding force proportional to sliding speed and spring constant $F_{slide} \propto Kvt$. An infinite spring constant can also be enforced for which the spring is omitted and the pull blocks are moved rigidly with a constant speed.

To limit the complexity of the friction behavior we want to consider systems without wear. To make sure that no wear is taking place for the sheet, we monitor the nearest neighbors for each atom throughout the simulation. At the initial timestep the three nearest neighbors, sitting at a distance 1.42 Å, of all graphene atoms are recorded. If any of these nearest neighbors exceeds a threshold distance of 4 Å, indicating a bond breakage, this is marked as a rupture and we halt the simulation. By conducting several test simulations involving high loads and high sliding speeds, we have visually confirmed that no wear occurs in the substrate, which demonstrates significantly greater wear resistance than the sheet. Therefore, we concluded that it was unnecessary to monitor the substrate for any signs of wear during the simulations.

5.3 Setting up the substrate

The substrate is created as a rectangular slab of silicon (Si). We create the initial configuration according to its crystalline structure given as a diamond cubic crystal with a lattice parameter $a_{\text{Si}} = 5.43 \text{ \AA}$. The default substrate thickness is chosen such that 9 layers of atoms appear (2 unit cells) corresponding to a thickness of

10.86 Å. The x-y dimensions are chosen to match the dimensions of the sheet. That is, we define a margin between the sheet edge and the substrate edge for the x- and y-direction respectively. Since we use periodic boundary conditions a too small margin would result in the sheet edges interacting with themselves through the boundary. The absolute lower limit for the margin choice is half the cut-off distance for the Tersoff potential, governing the graphene sheet interaction, at $R + D = 2.1$ Å. However, due to fluctuations in the sheet, we cannot set the margin too close to that limit. In addition, we need to consider the buckling of the sheet as it is stretched, which might cause an expansion in the x-direction for certain configurations. We choose an x-margin of 20 Å which provides $2 \cdot 20$ Å – 2.1 Å = 37.9 Å of additional spacing with respect to the absolute lower limit. By looking over the simulation result visually we confirm that this leaves more than enough room in the cases of most buckling. For the y-direction the rigid parts of the pull-blocks moves a certain distance based on the strain value exclusively, and we define the y-margin based on the remaining distance to the edge after stretching. However, as the sheet travels through the periodic boundaries in the y-direction when sliding, we want to add some additional spacing through the y-margin in order to let the substrate surface relax before interacting with the sheet a second time. We choose a y-margin of 15 Å for which the preferred sliding speed of 20 m/s = 0.2 Å/ps gives 150 ps of relaxation time between encounters with the sheet.

5.4 Setting up sheet

The sheet consists of graphene, which is a single layer of carbon atoms arranged in a hexagonal lattice structure. The bulk version of graphene is graphite and is a stacked structure of multiple graphene layers. We can describe the graphene 2D crystal structure in terms of its primitive lattice vectors \mathbf{a}_1 and \mathbf{a}_2 and a basis. The basis describes the atoms associated with each lattice site, and we populate the lattice by translating the basis by any linear combination of the lattice vectors

$$\mathbf{T}_{mn} = m\mathbf{a}_1 + n\mathbf{a}_2, \quad m, n \in \mathbb{N}.$$

For graphene, we have the primitive lattice vectors [135]

$$\begin{aligned} \mathbf{a}_1 &= a \left(\frac{\sqrt{3}}{2}, -\frac{1}{2} \right), & \mathbf{a}_2 &= a \left(\frac{\sqrt{3}}{2}, \frac{1}{2} \right), \\ |\mathbf{a}_1| &= |\mathbf{a}_2| = a = 2.46 \text{ Å}. \end{aligned} \quad (5.1)$$

Notice that we deliberately excluded the third coordinate as we only consider a single graphene layer and thus we do not have to consider the stacking structure of 3D graphite. The basis consists of two carbon atoms given as

$$\left\{ \left(0, 0 \right), \frac{a}{2} \left(\frac{1}{\sqrt{3}}, 1 \right) \right\}. \quad (5.2)$$

The crystal structure is visualized in Fig. 5.4. The hexagonal lattice structure makes for equal spacing between all pairs of atoms with an interatomic distance

$$\left\| \frac{a}{2} \left(\frac{1}{\sqrt{3}}, 1 \right) \right\| \approx 1.42 \text{ Å}.$$

5.4.1 Indexing

In order to describe the Kirigami cut patterns applied to the graphene sheet we require an indexing system that provides a unique representation of the atoms in the lattice. This allows us to represent the pattern as a binary matrix, where 0 denotes removed atoms and 1 denotes present atoms. We let the x-coordinate correspond to the so-called *armchair* direction of the sheet and the y-coordinate to the so-called *zigzag* direction. Notice that the x-coordinate will point to *zigzag* chains of atoms for which the starting point $(x, 0)$ is not evenly spaced as illustrated in Fig. 5.5. Other solutions might naturally involve the lattice vectors, but since these are used to translate between similar basis atoms it introduces an unfortunate duality as one would then need to include the basis atom of choice into the indexing system as well. Additionally, we want an indexing system that conserves the relative physical position of neighbors. That is, atom (i, j) should be in the proximity of $\{(i+1, j), (i-1, j), (i, j+1), (i, j-1)\}$. However, due to the hexagonal structure of the lattice, only three said neighbor indexes will be actual nearest neighbors in the lattice. While



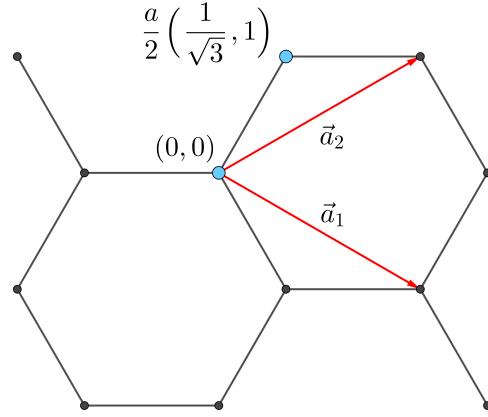


Figure 5.4: Illustration of the graphene crystal structure. The dots represent atom sites whereas blue dots denote the basis atoms (Eq. (5.2)). The red arrows denote the primitive lattice vectors (Eq. (5.1)).

$(i, j \pm 1)$ is always the nearest neighbor, the index of the nearest neighbor in the x-direction oscillates for each incrementing of the x- or y-coordinate. That is, the nearest neighbors NN are decided as

$$\begin{aligned} (i+j) \text{ is even} &\rightarrow \text{NN} = \{(i-1, j), (i, j+1), (i, j-1)\}, \\ (i+j) \text{ is odd} &\rightarrow \text{NN} = \{(i+1, j), (i, j+1), (i, j-1)\}. \end{aligned} \quad (5.3)$$

We can visually verify this by consulting Fig. 5.5, which shows that the nearest neighbor indexes depend on whether the atom is oriented to the left or right side in the zigzag chain.

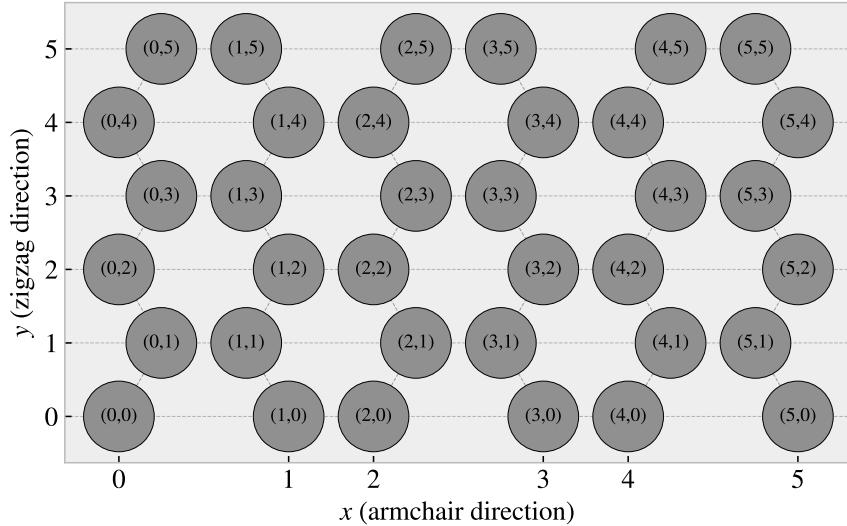


Figure 5.5: Illustration of the graphene atom site indexing. The x-coordinate increment along the armchair direction (pointing to zigzag chains) while the y-coordinate increment along the zigzag direction.

5.4.2 Removing atoms

To simplify the formulation of the cut patterns, we introduce the *center element* which is placed in each gap of the hexagonal honeycomb structure as shown in Fig. 5.6. These are not populated by any atoms but will serve as a reference for the numerical approaches to defining a cut pattern. The nearest neighbors of the center element alternate with position, similar to the atom site indexing. However, this time it is only dependent on the x-coordinate position. Each center element has six nearest neighbors, in a clockwise direction we can denote them: “up”, “upper right”, “lower right”, “down”, “lower left”, “upper left”. The “up” and “down” neighbors are always accessible as $(i, j \pm 1)$. However, for even i the $(i+1, j)$ index corresponds to the “lower right” neighbor while for odd i this corresponds to the “upper right” neighbor. This shifting applies for all left- or right-oriented neighbors and the full neighbor list is illustrated in Fig. 5.7. Analog to

the case of atom site indexing, we notice that the nearest neighbor indexes depend on whether the center element is oriented up or down on the armchair chain.

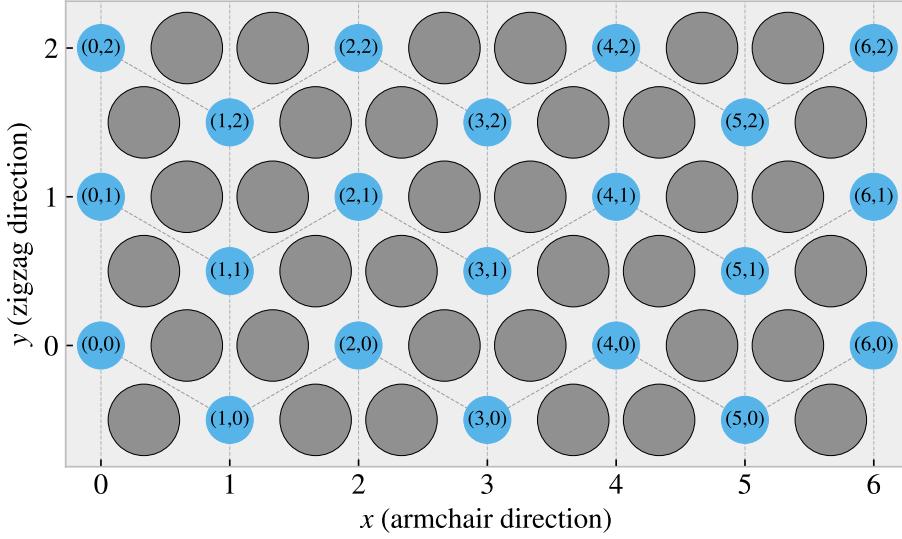


Figure 5.6: Illustration of the indexing for the introduced center elements depicted with blue circles placed in the gap of the honeycomb structure laid out by the graphene sheet atoms depicted with grey circles. The y-coordinate increment along the zigzag direction (pointing to armchair chains) while the x-coordinate increment along the armchair direction.

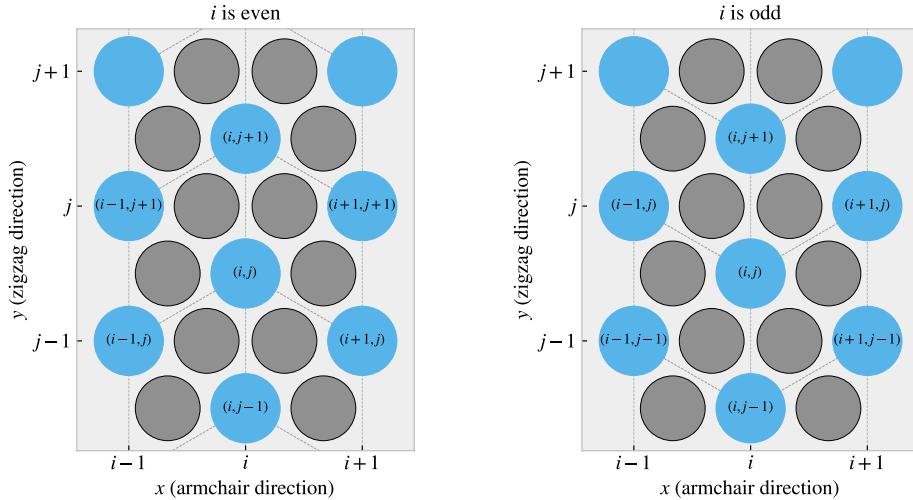


Figure 5.7: Illustration of the center element neighbor indexes for the case when the x-coordinate i is even (left) and i is odd (right).

We define a cut pattern by a connected path of center elements. As we walk between center elements through nearest neighbor walking we can remove atoms according to one of two rules

1. Remove intersecting atoms: We remove the pair of atoms placed directly in the path we are walking. That is, when jumping to the “up” center element we remove the two upper atoms located in the local hexagon of atoms. This method is sensitive to the order of the center elements in the path.
2. Remove all surrounding atoms: We simply remove all atoms in the local hexagon surrounding each center element. This method is independent of the ordering of center elements in the path.

We notice that removing atoms using either of these rules will not guarantee an injective, one-to-one, mapping. The first rule, being path dependent, will more often result in a unique result. However, for both methods, it is possible to construct two different paths leading to the same cut pattern as shown in the following



example:

$$\begin{aligned} \text{Path 1: } (i, j) &\rightarrow \underbrace{(i+1, j+1)}_{\text{upper right}} \rightarrow \underbrace{(i, j+1)}_{\text{up}} \rightarrow \underbrace{(i+1, j+2)}_{\text{upper right + up}} \rightarrow \underbrace{(i+1, j+1)}_{\text{upper right}} \\ \text{Path 2: } (i, j) &\rightarrow \underbrace{(i+1, j+1)}_{\text{upper right}} \rightarrow \underbrace{(i+1, j+2)}_{\text{upper right + up}} \rightarrow \underbrace{(i, j+1)}_{\text{up}} \end{aligned}$$

For the second rule, it is even more obvious that different paths can result in the same final pattern. For instance, if we incircle a center element completely there will be no surrounding atoms left to remove when jumping to that center element. This highlights the motivation for defining the atom-based indexing system which yields an injective mapping between the binary matrix and the graphene lattice cut pattern. However, using the center elements as a reference makes it easier to design the cut patterns since we can always go in one of the six directions defined by the center element neighbors. In contrast, the atom indexing system has alternating directions for its neighbors, which makes it more involved to define cut patterns.

5.5 Kirigami patterns

We propose a series of Kirigami-inspired cut patterns for the altering of the graphene sheet. We seek inspiration from macroscale patterns that showcase a considerable amount of out-of-plane buckling when stretched. We choose to imitate two different designs: 1) An alternating repeating series of perpendicular cuts as shown in Fig. 5.8a popularly used in studies of morphable metamaterials [136]. This pattern produces surface buckling with a tetrahedron (three-sided pyramid) shape when stretched. 2) A more intricate pattern shown in Fig. 5.8b which is used commercially by Scotch™ Cushion Lock™ [137] as protective wrap for items during shipping. This pattern buckles into a hexagonal honeycomb structure when stretched. In addition to the modeling of the so-called *Tetrahedron* and *Honeycomb* patterns, we also create a series of random walk patterns.

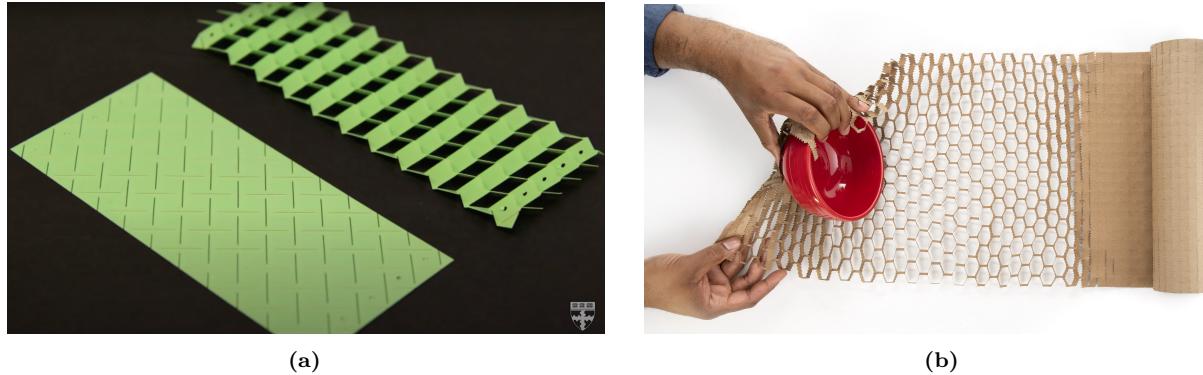


Figure 5.8: Macroscale kirigami cut patterns used as inspiration for the nanoscale implementation. (a) Tetrahedron: Alternating perpendicular cuts producing a tetrahedron-shaped surface buckling when stretched. Reproduced from [136]. (b) Honeycomb: Scotch™ Cushion Lock™ [137] producing a honeycomb-shaped surface buckling when stretched. Reproduced from [137].

5.5.1 Tetrahedron

The *Tetrahedron* pattern is defined in terms of center elements for which all atoms surrounding a given center element are removed. The pattern consists of two straight cuts, referred to as line 1 and line 2, that are arranged perpendicular to each other. The lines are positioned such that the center of one line aligns with the end of the other line, and with a given spacing in between (see Fig. 5.9). In order to achieve perpendicular cuts we cannot rely purely on the six center element directions corresponding to the center element neighbors which are spaced by 60° . We let line 1 run along the center elements in the direction of the “upper right” (and “lower left”) center elements, while line 2 goes in the direction between the “down” and “lower right” (“up” and “upper left”) center elements, corresponding to the direction $(1/\sqrt{3}, -1)$. We define variations of the pattern by the number of center elements L_1 and L_2 in line 1 and 2 respectively, together with the spacing between the lines d , as the tuple (L_1, L_2, d) . The pattern is constructed by translating the two lines to the whole sheet according to the spacing. Due to the alignment criteria of having one line point to the center of the other line, we can only allow an odd line length. Furthermore, in order to ensure that

each center element is translated to an i -index of similar odd or evenness, we must in practice require that $|L_2 - L_1| = 2, 6, 10, \dots$. Fig. 5.9 shows a visual representation of the pattern components for the $(7, 5, 2)$ pattern.

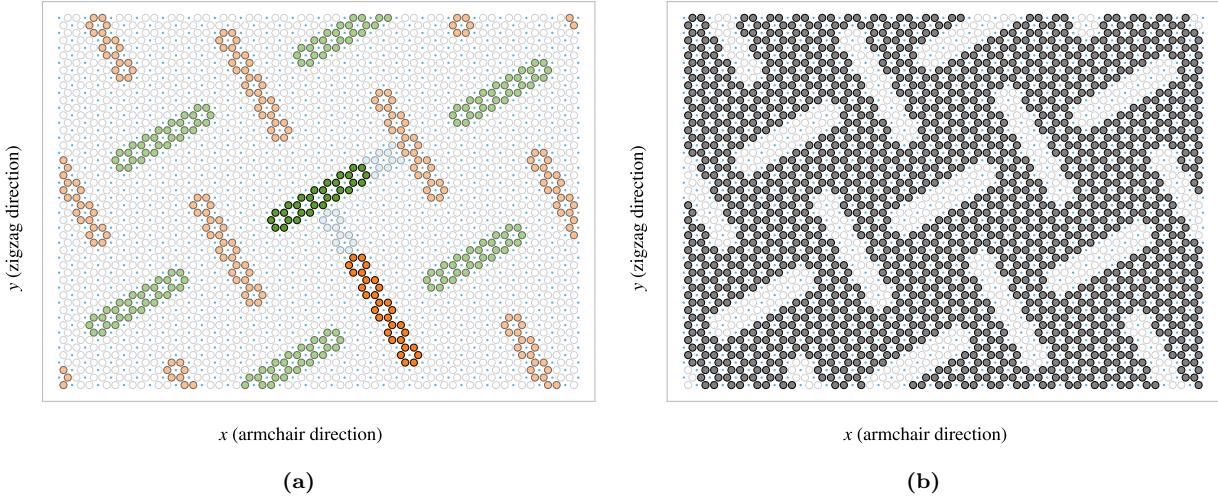


Figure 5.9: Visual representation of the Tetrahedron pattern consisting of two perpendicular lines, line 1 and line 2, of length L_1 and L_2 respectively, with spacing d . This example uses $(L_1, L_2, d) = (7, 5, 2)$ and a sheet matrix size 40×50 corresponding to 2000 atom sites and an approximate sheet size of $84 \times 77 \text{ \AA}$. The non-filled circles represent the possible atom site positions and the blue dots are the center elements. (a) Highlights the removed atoms in the pattern. Line 1 is shown in green and line 2 in orange, with lighter colors for the translated variations. The spacing is indicated in light blue. (b) The sheet after applying the cut pattern with grey circles denoting present atoms.

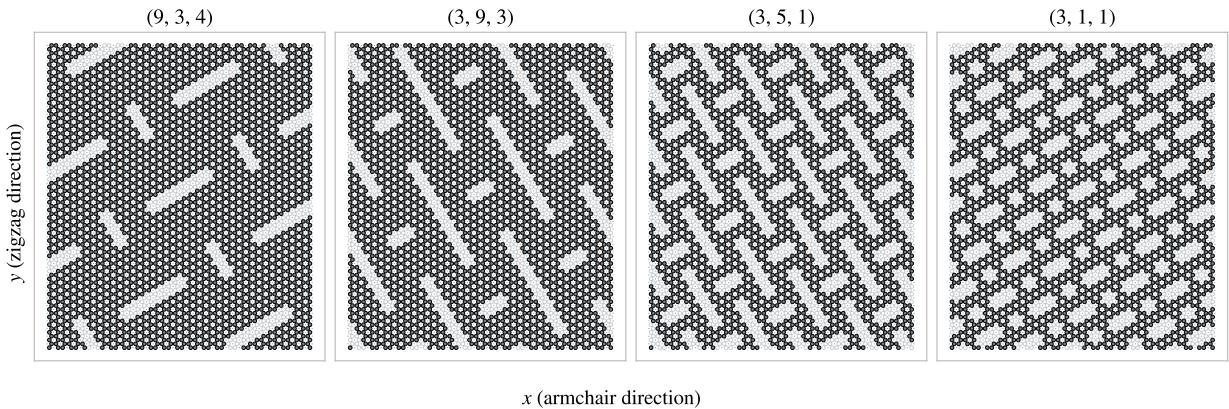


Figure 5.10: Example of different Tetrahedron cut pattern variations. The specific parameters are noted as titles and all of the patterns use the center of the sheet as a reference position. The circles in the figure represent atom sites, where grey-filled circles indicate the presence of atoms and transparent circles indicate removed atoms. The blue dots in the figure indicate the center elements. The sheet matrix is 40×80 corresponding to 3200 atom sites and an approximate sheet size of $84 \times 123 \text{ \AA}$.

In addition to the three parameters L_1, L_2, d , the pattern is also anchored to a reference point that describes the position of line 1 and line 2 before being translated to span the sheet. Due to the repeating structure of the pattern, there exist a small finite number of unique reference positions. For the pattern $(7, 5, 2)$ used as an example in Fig. 5.9, there are 140 unique reference points⁵. Some additional variation of the pattern is showcased in Fig. 5.10 each with a reference position at the center of the sheet. Note that a smaller sheet size than used in the simulations is used in both Fig. 5.9 and Fig. 5.10 for illustrative purposes.

⁵The general formula for calculating this number is rather complicated in comparison to its importance in this context. Therefore, we have omitted the formula and only provide the numerically backed result for the specific parameter set. The derivation of the formula was also deemed not to be rigorous enough.



5.5.2 Honeycomb

The *Honeycomb* pattern is defined, similarly to the Tetrahedron pattern, in terms of the center elements for which all surrounding atoms are removed. The Honeycomb pattern is built from a repeating series of cuts reminiscent of the Roman numeral one rotated by 90° (⊤). For a given spacing these are put next to each other in the x-direction, ⊤ ⊤ ⊤, to achieve a row where only a thin *bridge* in between is left to connect the sheet vertically in the y-direction. By placing multiple rows along the y-direction with alternating x-offset, we get the class of honeycomb patterns as visualized in Fig. 5.11. The pattern is described in terms of the parameters: (x-width, y-width, bridge thickness, bridge length) which is annotated in Fig. 5.11a with the parameters (2, 2, 1, 5) used as an example. Some additional variations of the pattern class are showcased in Fig. 5.12.

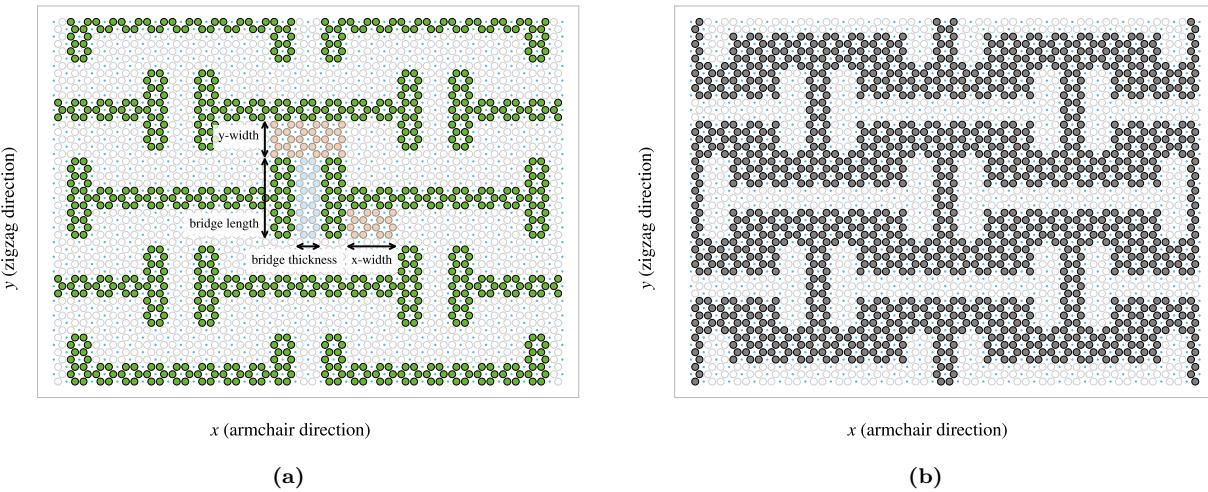


Figure 5.11: Visual representation of the Honeycomb pattern defined by the (x-width, y-width, bridge thickness, bridge length) parameters as annotated in panel (a). This example uses the parameters (2, 2, 1, 5) and a sheet matrix size 40×50 corresponding to 2000 atom sites and an approximate sheet size of $84 \times 77 \text{ \AA}$. The non-filled circles represent the possible atom site positions and the blue dots are the center elements. (a) Highlights the removed atoms in the pattern with annotations for the four defining parameters. (b) The sheet after applying the cut pattern with grey circles denoting present atoms.

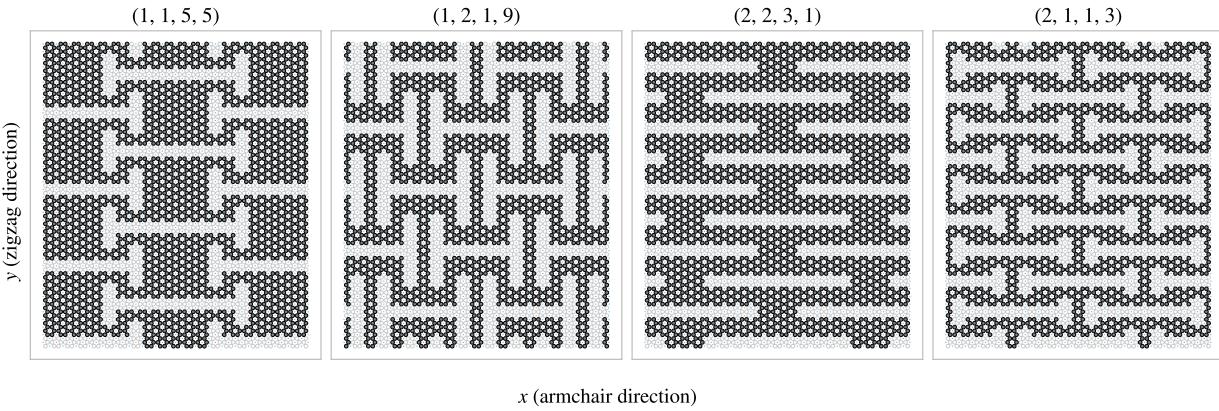


Figure 5.12: Example of different Honeycomb cut pattern variations. The specific parameters are noted as titles and all of the patterns use the center of the sheet as a reference position. The circles in the figure represent atom sites, where grey-filled circles indicate the presence of atoms and transparent circles indicate removed atoms. The blue dots in the figure indicate the center elements. The sheet matrix size is 40×80 corresponding to 3200 atom sites and an approximate sheet size of $84 \times 123 \text{ \AA}$.

5.5.3 Random walk

The random walk serves as a method for generating Kirigami patterns with randomized features. This approach is motivated by the aim of generating an ensemble of patterns that covers a larger region of the

configuration space than the more structured patterns mentioned earlier. This is considered to be important for the quality of the dataset used for machine learning. By this argument, a straightforward way to create random configurations could be achieved simply by random noise, either uniform or Gaussian. However, this would often leave the sheet detached with numerous non-connected atom clusters. Intuitively, we do not find this promising for the generation of large-scale structures which we hypothesize to be of interest. The random walk pattern generation is characterized by the parameters summarized in Table 5.3 which will be introduced throughout the following paragraphs.

Table 5.3: Parameters for the random walk generator.

| Parameter | Value | Description |
|----------------------|---------------------------------|--|
| Num. walkers (M) | Integer ≥ 1 | Number of random walks to be initiated on the sheet (one at a time). |
| Max. steps (S) | Integer ≥ 1 | The maximum steps allowed for any random walker. |
| Min. distance | Integer ≥ 0 | The minimum distance required between any future paths and the previous paths in terms of the shortest walking distance in between. |
| Bias | (direction, strength ≥ 0) | Bias direction and strength defining the discrete probability for the choice of the next site. |
| Connection | Atoms / Center elements | Whether to walk between atom sites or center elements removing all adjacent atoms. |
| Avoid invalid | True/False | Whether to remove already visited sites from the neighbor list before picking the next site. This prevents jumping to already visited sites and lowers the likelihood of early termination. |
| Stay or break | $p = [0, 1]$ | Probability that the walker will maintain its direction for the next step. |
| Periodic | True/False | Whether to use periodic boundary conditions on all four sides. |
| Avoid clustering | Integer ≥ 0 | Amount of times to restart the whole random walk generation in order to arrive at a non-detached configuration. If no valid configuration is reached after this number of attempts, the non-spanning clusters are removed. |
| RN6 | True/False | Randomly change the bias direction between the deployment of each random walker to one of the six center element directions. |
| Grid start | True/False | The option to have the random walkers start in an evenly spaced grid. |
| Centering | True/False | Relocate the path of a random walk after termination such that the path center of mass gets closer to the starting point (without violating the rules regarding already visited sites). |

5.5.3.1 Fundamentals

For an uncut sheet, we deploy M random walkers, one at a time, and let them walk for a maximum number of S steps. We can either let the walker travel between atom sites, removing the atoms in the path as it goes, or between the center elements, removing all surrounding atoms. This is managed by setting the *connection* parameter to either *atom* or *center elements*. The method of removing only the intersecting atoms between center elements was also incorporated, but we ended up not using it due to plenty of other interesting options. Nonetheless, we will always remove a site once visited such that the walker itself, or any other walkers, cannot use this site again. This leads to a self-avoiding random walk, meaning that the walker does not cross its own path. However, it furthermore constrains the walkers to avoid the path taken by any previous walker, and thus we might denote this property as “other avoiding”. By default, the walker has an equal chance of choosing any of its adjacent neighbors for the next step, i.e. we draw the next step from a discrete uniform distribution. Optionally, we can use periodic boundary conditions, by setting the parameter *periodic* to true, allowing neighboring sites to be connected through the edge in both the x and y-direction. When



traveling on atom sites this ensures that we have three neighbor options for the next step while traveling on the center elements this gives six neighbor options. If the walker happens to arrive at an already visited site the walk is terminated early. Optionally, we can choose to remove any neighboring sites already visited from the neighbor list and choose uniformly between the remaining options instead. This is done by setting the parameter *avoid invalid* to true. This prolongs the walking distance, but the walker is still able to find itself in a situation where no neighboring sites are available. Note that the walker is not allowed to backtrack its own path either, and thus in such a case the walk will be terminated despite the setting of *avoid invalid*.

5.5.3.2 Spacing of walking paths

To control the spacing between the paths of the various walkers, we implement a so-called *minimum distance* parameter, taking integer values ≥ 0 . This parameter describes the minimum spacing required between paths in terms of the least amount of walking steps. When a walker has ended its walk, either by early termination or hitting the maximum limits of steps, all sites within walking distance corresponding to the minimum distance parameter are marked as visited, although they are not removed from the sheet. This prevents any subsequent walkers to visit those sites in their walk according to the general behavior introduced in the previous paragraph. In practice, this is done through a recursive algorithm as described in Algorithm 1. For a given path the function *walk_distance()* is called with the input being a list of all sites in the given paths. The function gathers all the neighbors of each site, regardless of their state on the sheet. It then calls itself recursively using this neighbor list as input, while incrementing a distance counter that is also passed along as an argument. This results in an expansion along all possible outgoing paths from the initial path of interest. Once the distance limit is reached, the function returns the final neighbor lists, which are then accumulated into a final output. This output corresponds to a list of all sites within the minimum distance to the path.

Algorithm 1 Recursive algorithm implemented as a class method of the random walk generator. For a given path input it flags all sites within a distance given by the class attribute *self.min_dis*.

Require: *self.min_dis* > 0

```

1: function WALK_DISTANCE(self, input, dis = 0, pre = [ ])
2:   new_neigh  $\leftarrow$  [ ]                                ▷ Initialize list for new neighbors
3:   for site in input do
4:     neigh  $\leftarrow$  get_neighboring_sites(site)           ▷ Get surrounding neighbors
5:     for n in neigh do
6:       if (n not in pre) and (n not in new_neigh) then      ▷ If not already added
7:         AddItem(new_neigh, n)                           ▷ then add the site
8:       end if
9:     end for
10:    end for                                         ▷ Increment distance counter
11:    dis += 1                                         ▷ Max limit hit
12:    if dis  $\geq$  self.min_dis then
13:      return input + new_neigh
14:    else                                              ▷ Start a new walk from each of the neighboring sites
15:      pre  $\leftarrow$  input
16:      return pre + self.walk_distance(new_neigh, dis, pre)
17:    end if
18:  end function
```

5.5.3.3 Bias

We provide the option to perform a biased random walk by specifying the bias parameter, which consists of a direction and a strength. To achieve this, we model each step of the walk analog to a system in the canonical ensemble under the influence of an external force \mathbf{F} representing the bias. For such a system each microstate i , corresponding to the sites in the neighbor list, has the associated probability p_i given by the Gibbs–Boltzmann distribution

$$p_i = \frac{1}{Z} e^{-\beta E_i}, \quad Z = \sum_i e^{-\beta E_i},$$

where Z is the canonical partition function, $\beta = 1/k_B T$ for the boltzmann constant k_B and temperature T , and E_i the energy of site i . We model the energy of each site as the work required to move there. For a step

so the energy becomes $E_i = -\mathbf{s} \cdot \mathbf{F}$, where the sign is chosen such that the energy (difference) is negative when moving along the bias, analogous to an energy gain by moving there. Due to the symmetry of the random walk sites, both for the atom sites and the center elements, the step length to neighboring sites will always be equal. By defining the bias strength $B = \beta |\mathbf{F}| |\mathbf{s}|$ we get that the probability for jumping to site i is

$$p_i = \frac{1}{Z} e^{B\hat{\mathbf{s}} \cdot \hat{\mathbf{F}}} \propto e^{B\hat{\mathbf{s}} \cdot \hat{\mathbf{F}}},$$

where the hat denotes the unit vector. The bias strength B then captures the opposing effects of the magnitude of the external force and the temperature of the system since $B \propto |\mathbf{F}|/T$. We notice that $\hat{\mathbf{s}} \cdot \hat{\mathbf{F}} = \cos(\theta)$ for the angle θ between the step and bias direction. This shows that the bias will have the biggest positive contribution to the probability when the step direction is fully aligned with the bias direction ($\theta = 0$), have no contribution for orthogonal directions ($\theta = \pm\pi/2$) and the biggest negative contribution when the directions are antiparallel ($\theta = \pi$). The partition function serves as a normalization constant. Thus, numerically we can enforce this simply by setting $Z = 1$ at first, calculating p_i , and then normalizing the result at the final stage as a division by the sum of all p_i . In the numerical implementation, we then pick the next step by the weighted discrete probability distribution p_i . In Fig. 5.13 we have illustrated how a bias of different strengths impacts the probability distribution for a random walk between center elements. We can visually confirm that the bias will favor sites that lie close to the bias direction. This preference is more distinct at high bias strengths while at low strength $B \rightarrow 0$ we get a uniform distribution that aligns with the default unbiased random walk.

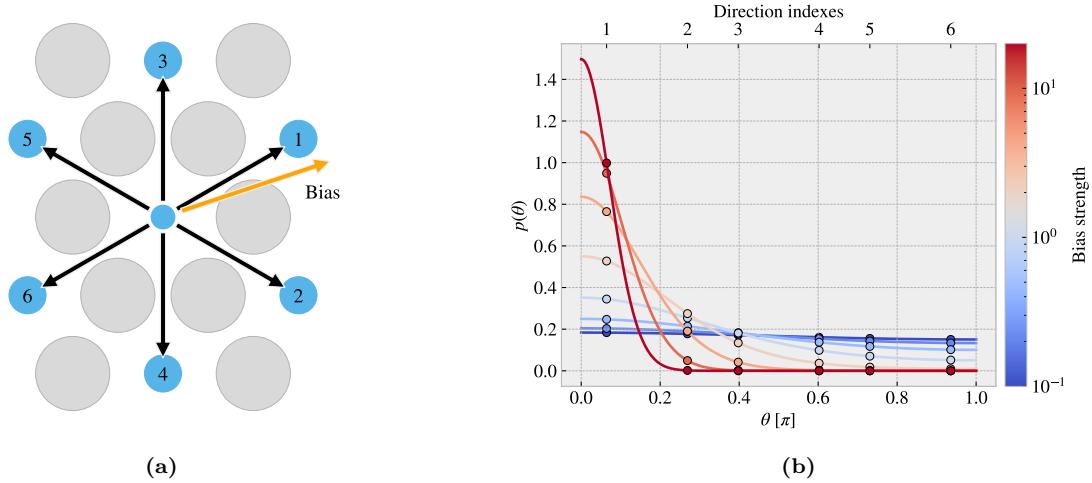


Figure 5.13: Illustration of the probability distribution for the various step directions during a biased random walk between center elements. (a) The possible step directions are represented by black arrows that point towards the neighboring center elements depicted as blue circles. The bias direction is denoted by an orange arrow, and the numbering indicates the most probable direction (1) towards the least probable direction (6). The atom sites are marked as grey circles for reference. (b) The probability distribution as a function of the angle θ between the step direction and the bias direction. The distribution is normalized according to the discrete probabilities marked with dots for which the continuous line simply highlights the shape of the distribution. The direction indexes correspond to the numbering on panel (a). The color map indicates different strengths of the bias.

5.5.3.4 Stay or break

The *stay or break* parameter defines the probability p_{stay} that the walker will keep its direction or otherwise break into a different direction by probability $1 - p_{\text{stay}}$. We implement this by altering the discrete probability used for the choice of the next step. We manually set the probability to p_{stay} for the site corresponding to a continuation in the same direction and renormalize the distribution. This allows us to perform a biased random walk in combination with a preference for keeping direction. For the center element walk it is trivial to determine which of the neighbor directions correspond to a continuation of direction based on the last visited site. However, for an atom site walk, it is not possible to follow the same direction in a straight line due to the hexagonal layout of the lattice. We recall that the nearest atom neighbor indexes alternate for each increment in the x or y index (see Eq. (5.3)) which corresponds to the alternating neighbor directions



D as

$$(i+j) \text{ is even} \rightarrow D = \left\{ \frac{a}{2} \left(\frac{-2}{\sqrt{3}}, 0 \right), \frac{a}{2} \left(\frac{1}{\sqrt{3}}, 1 \right), \frac{a}{2} \left(\frac{1}{\sqrt{3}}, -1 \right) \right\},$$

$$(i+j) \text{ is odd} \rightarrow D = \left\{ \frac{a}{2} \left(\frac{2}{\sqrt{3}}, 0 \right), \frac{a}{2} \left(\frac{-1}{\sqrt{3}}, 1 \right), \frac{a}{2} \left(\frac{-1}{\sqrt{3}}, -1 \right) \right\}.$$

One way to mitigate this issue is to use the six directions from the center element walk as the common direction to “stay or break” from. As showcased in Fig. 5.14, for each center element direction (black arrows) there are two possible atom site directions (red and orange arrows) that are equally close to the center element direction. The red and orange arrows represent $(i+j)$ being even or odd respectively, and we notice that these appear in pairs such that we can always uniquely determine which of the atom directions is closest to the center element direction. Following this idea we can map each center direction to an atom direction depending on the even or oddness of the position. For $p_{\text{stay}} = 1$ this results in a guaranteed zigzag motion along the center element direction that it happens to start on.

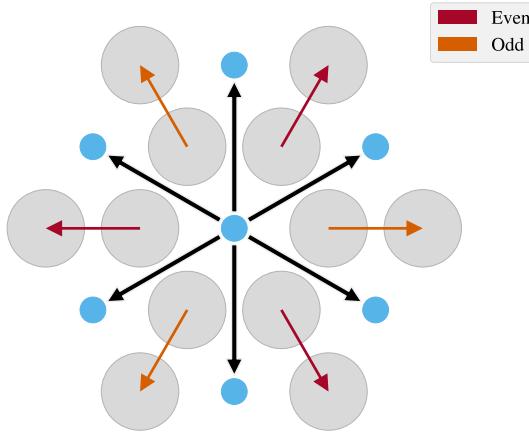


Figure 5.14: Visualization of the center element directions (black arrows) connecting the center elements depicted as blue dots, and the atom site directions (red and orange arrows) connection the atom sites depicted as grey circles. The red arrows correspond to the case where the sum of the atom site indexes (i, j) is even, while the orange arrows correspond to the case where the sum is odd. Notably, for each case, there is always one atom site direction that is uniquely closest to a given center element direction.

The *stay or break* feature is still subject to previously defined rules. For instance, in the case where the preferred site is not available, the walker will either terminate when going there, or the preferred site is removed from the neighbor list when *avoid invalid* is set to true. In the latter case, the walker will be forced to break out of its direction and follow the new direction that it happens to choose.

5.5.3.5 Deployment schemes

By default, each random walker is given a uniform random starting point among the non-visited available sites left on the sheet. This includes any modifications in relation to the minimum distance parameter. By setting the *grid start* parameter to true, the starting points are instead predefined on an evenly spaced grid. That is, the sheet is subdivided into the least amount of squares that will accommodate space for each starting point. {1} walker leads to a 1×1 partition, {2, 3, 4} walkers lead to a 2×2 partition, {5, 6, 7, 8, 9} walker lead to a 3×3 partition and so on. For each partition square, the starting point is placed as centrally as possible. The lower left partition square is then chosen as a default starting place for the first walker and the remaining sites are filled according to the order that maximizes the minimum distance between a new starting point and the ones already occupied⁶. An example of the deployment is shown in Fig. 5.15. Notice, that if the planned grid start position is made invalid before deployment, it is skipped.

The *centering* parameter lets us relocate the path of the random walker such that the path center of mass gets closer to the starting point. When set to true, the path is moved in the direction defined from the center of mass toward the starting point for which the closest valid relocation on the direct translation line is chosen. This can be used in combination with the *grid start* and the *bias parameter* to make rather ordered

⁶In hindsight, we realize that it would have been less biased to choose a random partition square as the starting one, but we do not consider this to be of great importance for the usage of this feature in the dataset.

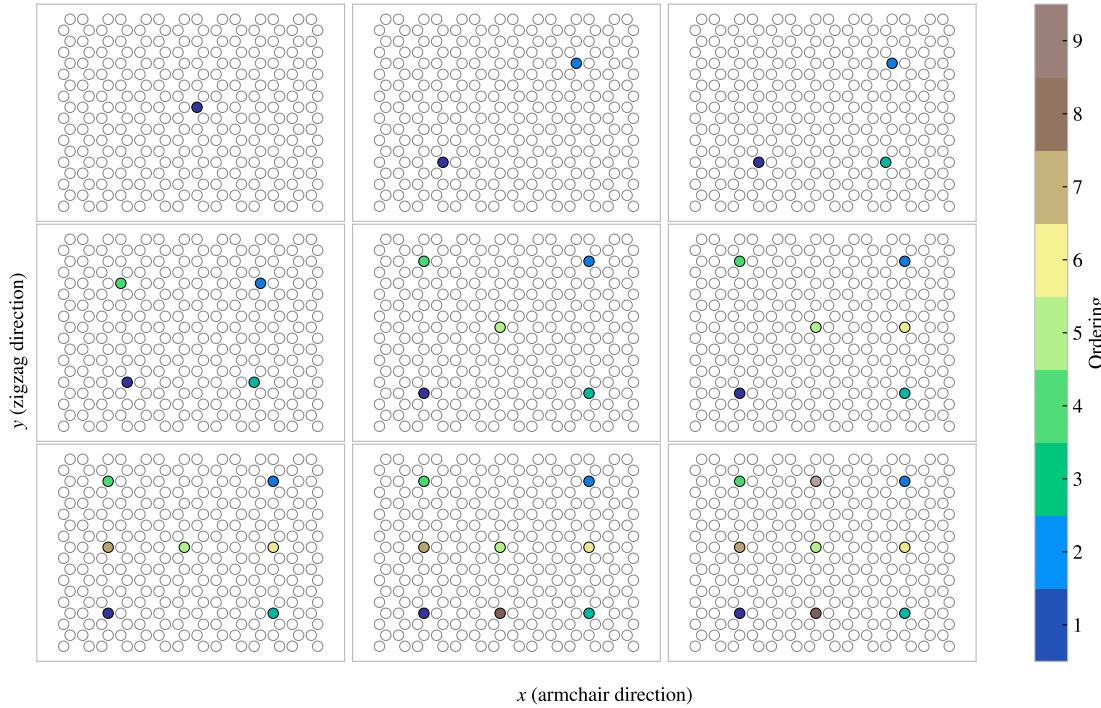


Figure 5.15: The figure illustrates the distribution of starting points when the *grid start* parameter is set to true for a 14×18 sheet, for varying numbers of deployed random walkers ranging from 1 to 9. The color map is used to indicate the order in which the walkers were deployed.

configurations. In addition, the *RN6* parameter can be used to update the bias direction to one of the six center element directions for each new walker deployed. This lets us create more organized configurations like the one shown in Fig. 5.16b.

5.5.3.6 Validity

The simulation procedure requires the sheet to be fully attached, non ruptured, which can be summarized as the following requirements.

1. There exists only a single cluster on the sheet. We define a cluster as the set of atoms which can all be reached through nearest neighbor walking on the cluster.
2. The cluster of atoms is spanning the sheet in the y-direction. This means that there exists at least one path through nearest neighbor walking that connects the bottom and the top of the sheet. This is required because the sheet must be attached to the pull blocks in the simulation.

In order to accommodate these requirements we count the number of clusters and search for a spanning cluster after all walkers have terminated. If the requirements are not met we simply rerun the random walk from scratch. This is done according to the *avoid clustering* parameter which takes integer values corresponding to the number of times to repeat this process. If the requirements are not met during any of those reruns the non-spanning clusters are simply removed. In the case of no spanning cluster, the configuration is skipped. This crude scheme was later reinvented as a more refined repair scheme that alters the sheet with the intention of performing the least amount of changes, addition or subtraction of atoms, in order to meet the attachment requirements. This was done as a part of the accelerated search procedure and hence it was not utilized in the creation of the random walk dataset. However, we give a brief description of the algorithm here:

1. Find all clusters and rank them in descending order of size, such that the largest cluster is labeled as cluster 1.
2. Deploy walkers from the edges of the smallest cluster, allowing them to walk in all possible directions similar to what was done with Algorithm 1. Here the allowed walking distance is defined as the number of atoms within the starting cluster.



3. If a walker from the starting cluster reaches another cluster within the allowed walking distance, these two clusters are connected through the walking path. Otherwise, the starting cluster is deleted. This decision is based on the consideration that the number of atoms required to connect the clusters should not exceed the number of atoms removed if the starting cluster is deleted. However, if the starting cluster is the last cluster connected to the top or bottom edge, an unlimited number of walks are allowed to ensure the creation of a spanning cluster.
4. If there is only one cluster remaining on the sheet, the repair procedure is complete. If there are still multiple clusters, then repeat the steps from 1. Note, that if no spanning cluster was ever present, we expand the final cluster until a spanning connection can be established.

5.5.3.7 Random walk examples

Some examples of the random walk patterns are illustrated in Fig. 5.16 with corresponding parameters in Table 5.4.

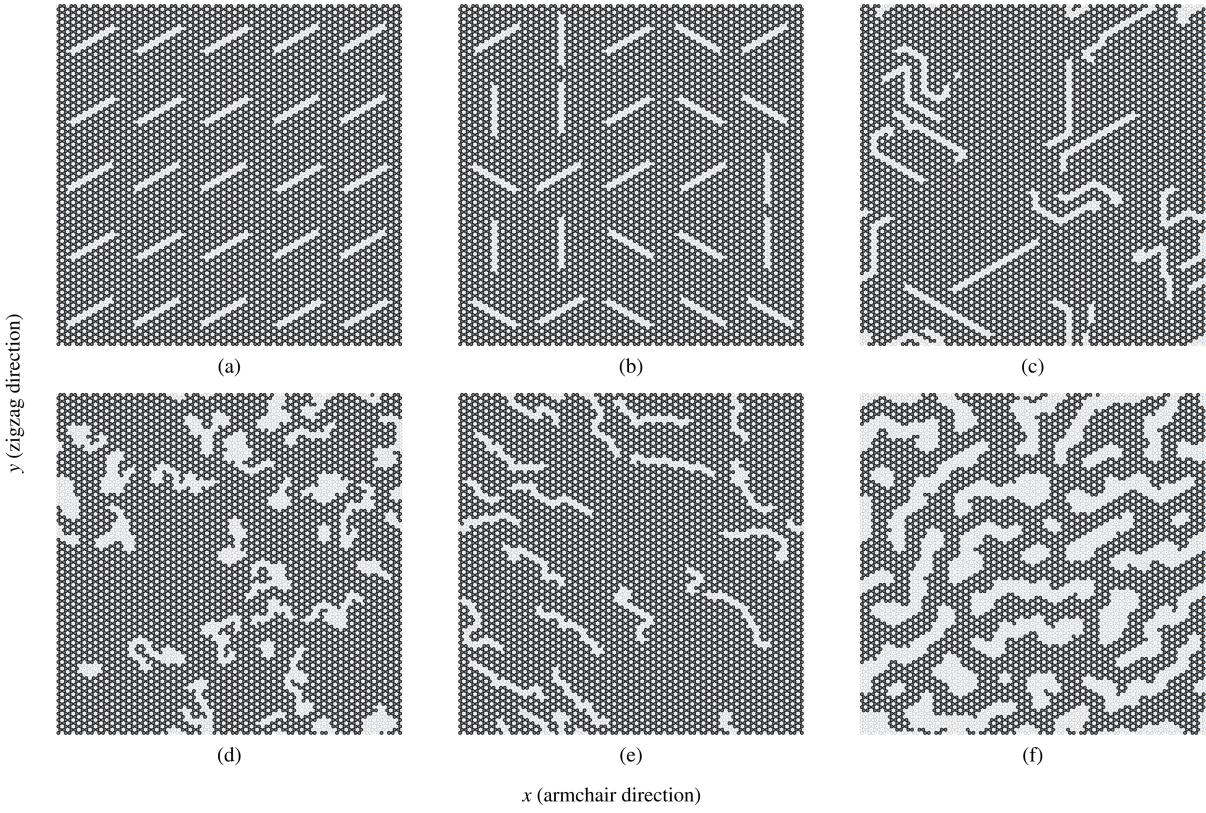


Figure 5.16: Example of different Random walk cut pattern variations. The specific parameters are given in Table 5.4. The circles in the figure represent atom sites, where grey-filled circles indicate the presence of atoms and transparent circles indicate removed atoms. The sheet matrix size is 62×106 corresponding to the full-size system used in the MD simulations with 6572 atom sites and an approximate sheet size of $130 \times 163 \text{ \AA}$.

Table 5.4: Parameters for the random walk patterns shown in Fig. 5.16.

| Fig. num. | (a) | (b) | (c) | (d) | (e) | (f) |
|------------------|---|---|-------|-------|---|--|
| Num. walkers | 25 | 25 | 20 | 30 | 20 | 32 |
| Max. Steps | 15 | 15 | 30 | 40 | 30 | 30 |
| Min. distance | 0 | 0 | 4 | 4 | 4 | 4 |
| Bias | $\begin{pmatrix} \text{upper right} \\ 100 \end{pmatrix}$ | $\begin{pmatrix} \text{upper right} \\ 100 \end{pmatrix}$ | None | None | $\begin{pmatrix} \text{lower right} \\ 1.2 \end{pmatrix}$ | $\begin{pmatrix} \text{lower left} \\ 1.2 \end{pmatrix}$ |
| Connection | Atoms | Atoms | Atoms | Atoms | Atoms | Center elements |
| Avoid invalid | False | False | True | True | True | True |
| Stay or break | 0 | 0 | 0.9 | 0 | 0 | 0 |
| Periodic | True | True | True | True | True | True |
| Avoid clustering | 10 | 10 | 10 | 10 | 10 | 10 |
| RN6 | False | True | True | False | False | False |
| Grid start | True | True | False | False | False | False |
| Centering | True | True | False | False | False | False |



Chapter 6

Pilot study

In this chapter, we begin by analyzing the frictional properties of our system. Initially, we examine the results for a non-cut sheet to establish suitable metrics for a numerical evaluation of friction and validate our parameter choices. This provides a basis for the following study where we explore the frictional properties of the Tetrahedron and Honeycomb Kirigami patterns. We conduct a more thorough investigation of the friction dependencies to temperature, sliding speed, spring constant, and timestep. Finally, we investigate the kinetic friction across all three configurations when subjected to strain, including an examination of the relationship to contact area and the friction-load curves.

6.1 Friction simulation parameters

The MD simulations we will carry out to measure friction are governed by a small set of parameters. Since we aim to develop a machine-learning model, it is necessary to standardize these parameters. Therefore, we keep the majority of the parameters constant and only modify a small subset of them, which includes sheet configuration, strain, and load. Instead of starting with the parameter selection process, we first state the final choice in Table 6.1. Due to the great number of parameters, we did not make an exhaustive search of all parameters before deciding on the final choice. Instead, we have taken a basis in parameters used in similar friction simulations [78, 79, 84–86] and adjusted accordingly to the aim of getting stable measurements and reducing computation time where possible. Parameters such as initial relaxation time, pauses and strain speed are chosen mainly from the results of initial stability tests. That is, we visually verify that the system is close to an equilibrium state and that it does not carry momentum before going to the next step in the numerical procedure. The sheet and pull block sizes are chosen with a consideration of the balance between Kirigami design options and computational resources. The scan direction is selected to be parallel with the connecting line between the pull blocks. This choice is made primarily to minimize the complexity of the motion, as it is hypothesized that other scan directions might cause the center of the sheet to lag behind in the sliding motion and produce a slight side flexion. The remaining parameters: Temperature T , sliding speed v , spring constant K , normal load F_N , timestep dt and sliding distance have mainly been chosen because the friction output remains relatively stable with moderate perturbations around these default values. However, we will explain the parameter choices in more detail throughout this chapter. Note that the default values in Table 6.1 will be used when nothing else is stated explicitly.

6.2 Force traces

We begin by assessing the friction force traces, i.e. force vs. time curves, for a single friction simulation using the default parameters shown in Table 6.1 for a non-cut sheet with no strain applied and a normal load of 1 nN.

6.2.1 Force oscillations

We evaluate the friction force as the force acting on the sheet from the substrate. We consider initially the force component F_{\parallel} parallel to the sliding direction shown in Fig. 6.1. We use a sample rate of 10 ps^{-1} corresponding to one sample for each 100 timesteps, with each sample being the mean value of the preceding 100 timesteps. We observe immediately that the data carries oscillations on different time scales which match our general expectations for sliding involving periodic surfaces. By applying a Savgol filter to the data with a polynomial order of 5 and a window length of 150 data points (corresponding to a sliding



Table 6.1: Parameters involved in the numerical MD simulation for measuring friction. The default values correspond to the final choice used for the ML dataset for which the shaded cells denote the parameters varied.

| Parameter | Default value | Description |
|-------------------------|--|--|
| T | 300 K | Temperature of the system. |
| v | 20 m/s | Sliding speed for the sheet. |
| K | ∞ | Spring constant for the coupling between the virtual atom and the sheet pull blocks. |
| Scan direction | $(x, y) = (0, 1)$ (zigzag direction) | The direction of sliding. |
| Sheet configuration | Contiguous | Binary mapping describing which atoms are removed (0) and which are still present (1) in the graphene sheet. |
| Strain amount | [0, rupture] | The ratio of change in length to the original length. |
| F_N | [0.1, 10] nN | Applied normal load to the pull blocks. |
| dt | 1 fs | MD integration timestep. |
| Initial relaxation time | 15 ps | Initial relaxation time before straining. |
| Pauses | 5 ps | Relaxation pauses after strain, and during the normal load phase (before sliding). |
| Strain Speed | 0.01 ps^{-1} | The rate of straining for the sheet. |
| Slide distance | 400 \AA | How far the sheet is slid. |
| Sheet size | $130.029 \times 163.219 \text{ \AA}$ | Spatial 2D size of the sheet. |
| Pull block size | $2 \times 130.029 \times 15.183 \text{ \AA}$ | Spatial 2D size of the pull blocks. |

distance of 3 \AA or a time window of 15 ps) we can qualitatively point out at least two different frequencies of oscillation. During the first 10 \AA of sliding, seen in Fig. 6.1a, we see roughly three waves on the Savgol filter corresponding to a relatively high frequency, while for the duration of 100 \AA of sliding, seen in Fig. 6.1b, the same Savgol filter reveals a lower frequency on top, creating the visual pattern of a wavepacket. The data does not indicate signs of stick-slip behavior as otherwise found in other studies, e.g. by Zhu and Li [85] for graphene on gold, who saw a more typical saw tooth shape in the force trace. Besides the difference in the substrate material, using gold instead of silicon, they used a lower sliding speed of 10 m/s and a soft spring of $K = 10 \text{ N/m}$. By adopting these parameters we get a slightly different force trace behavior as shown in Fig. 6.1c and Fig. 6.1d. This parameter change results in a loss of symmetry in the force oscillations, but it still does not produce any significant discontinuities in the trace. By keeping the spring constant at $K = 10 \text{ N/m}$ and lowering the sliding speed further down to 1 m/s, we achieve a more considerable stick-slip behavior as shown in Fig. 6.1e and Fig. 6.1f. Considering all three simulations, we might classify the results from the default settings, $K = \infty$, $v = 20 \text{ m/s}$, as smooth sliding, $K = 10 \text{ N/m}$, $v = 10 \text{ m/s}$, as a transition phase with possible occasional slipping, and $K = 10 \text{ N/m}$, $v = 1 \text{ m/s}$ as a more typical stick-slip behaviour. This confirms the qualitative observation that stick-slip behavior is suppressed with stiff springs [82] and high sliding speed [84]. Having a low sliding speed comes with a high computational cost which is the reason that we choose a relatively high sliding speed of 20 m/s. The choice of an infinite spring constant is related to the stability of the measurements and is discussed later in this chapter.

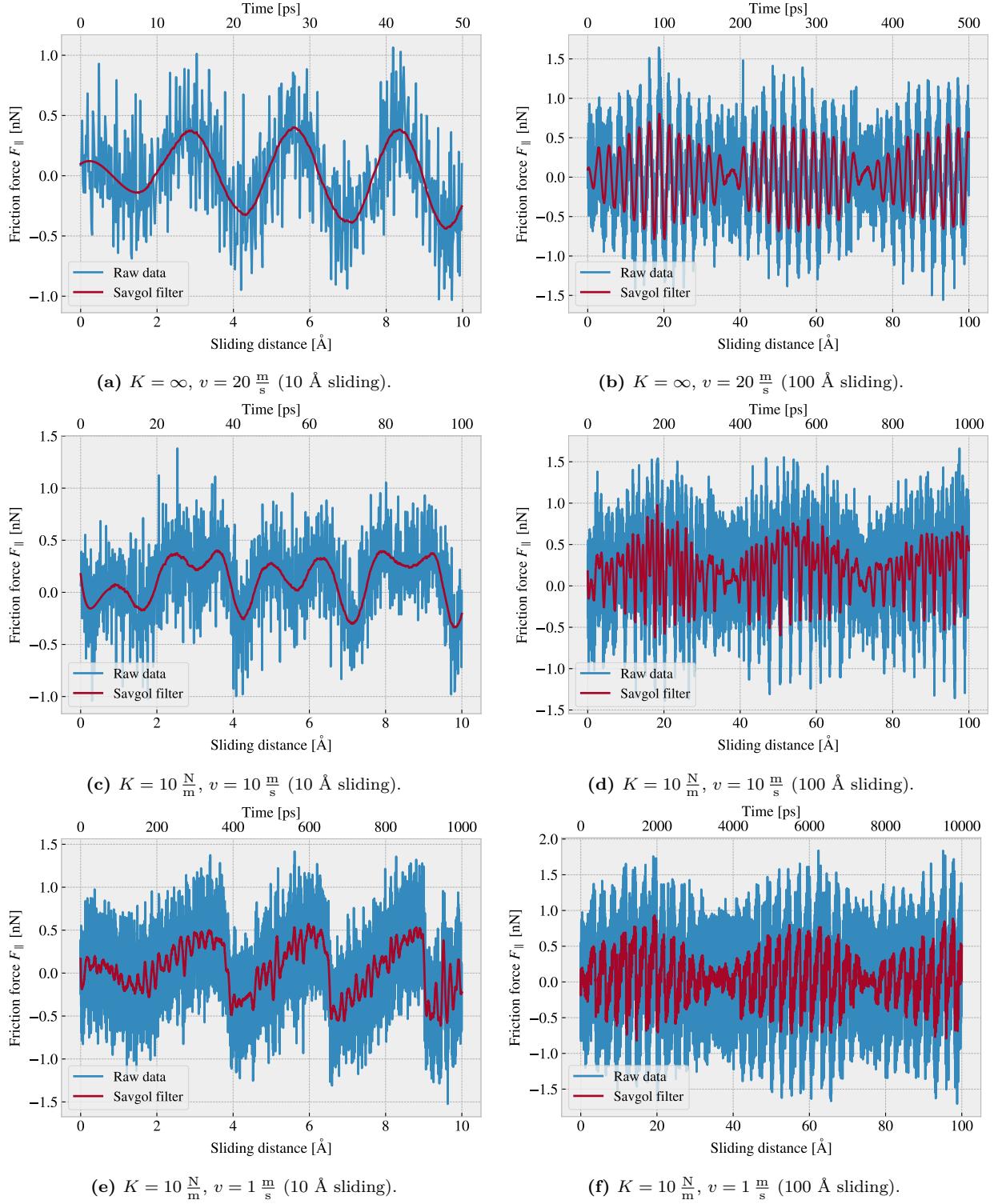


Figure 6.1: Force traces of the friction force component F_{\parallel} parallel to the sliding direction acting from the substrate on the full sheet. The force traces are plotted against the sliding distance (lower x-axis) and the corresponding sliding time (upper x-axis). The sliding distance is measured by the displacement of the virtual atom tethering the sheet. The red line represents a Savgol filter with polynomial order 5 and a window length of 150 data points (corresponding to a sliding distance of 3 \AA or a time window of 15 ps). Each row, (a,b), (c,d), (e,f), represents a different choice of the spring constant K and sliding speed v , while the columns show the same result for two different time scales. The default settings are represented in figures (a) and (b).



By performing a Fourier Transform on the data, using the default parameters, we can quantify the leading frequencies observed in Fig. 6.1a and Fig. 6.1b. The Fourier transform is shown in Fig. 6.2a, and by plotting the two most dominant frequencies $f_1 = 0.0074 \text{ ps}^{-1}$ and $f_2 = 0.0079 \text{ ps}^{-1}$ as a sine sum, $\sin(2\pi f_1) + \sin(2\pi f_2)$, we find a qualitatively convincing fit to the observed wavepacket shape as seen in Fig. 6.2b. We convert the frequencies according to that of a wavepacket. By using the trigonometric identity

$$\begin{aligned}\sin(a+b) &= \sin(a)\cos(b) + \cos(a)\sin(b), \\ \sin(a-b) &= \sin(a)\cos(b) - \cos(a)\sin(b),\end{aligned}$$

and decomposing the frequencies as $f_1 = a - b$, $f_2 = a + b$, we can rewrite the sine sum as the sinusoidal product

$$\begin{aligned}\sin(2\pi f_1) + \sin(2\pi f_2) &= \sin(2\pi(a-b)) + \sin(2\pi(a+b)) \\ &= \sin(2\pi a)\cos(2\pi b) - \cos(2\pi a)\sin(2\pi b) + \sin(2\pi a)\cos(2\pi b) + \cos(2\pi a)\sin(2\pi b) \\ &= 2\sin(2\pi a)\cos(2\pi b),\end{aligned}$$

with

$$\begin{aligned}a = \frac{f_1 + f_2}{2} &= 0.0763 \pm 0.0005 \text{ ps}^{-1}, & b = \frac{f_2 - f_1}{2} &= 0.0028 \pm 0.0005 \text{ ps}^{-1}, \\ &= 0.382 \pm 0.003 \text{ \AA}^{-1}, & &= 0.014 \pm 0.003 \text{ \AA}^{-1}.\end{aligned}$$

In the latter transition, we have denoted the frequency with respect to the sliding distance by considering the default sliding speed of $20 \text{ m/s} = 0.2 \text{ \AA}/\text{ps}$. This makes us recognize the high oscillation frequency as a and the low frequency as b . The faster one has a period of $T_a = 2.62 \pm 0.02 \text{ \AA}^7$ which corresponds well with the magnitude of the lattice spacing as expected theoretically, and especially that of graphene with a lattice constant of 2.46 \AA . The longer period $T_b = 71 \pm 15 \text{ \AA}$ is not obviously explained. We notice a similarly long period oscillation for all three cases Fig. 6.1b, Fig. 6.1d and Fig. 6.1f, and thus we have no reason to believe that this is dependent on the stick-slip behavior. The initial build-up in friction force is reminiscent of friction strengthening, which is often reported [77, 79], but the periodicity goes against this idea. Instead, we might attribute it to a phonon resonance effect which might be further affected by the fact that we use periodic boundary conditions for a finite substrate.

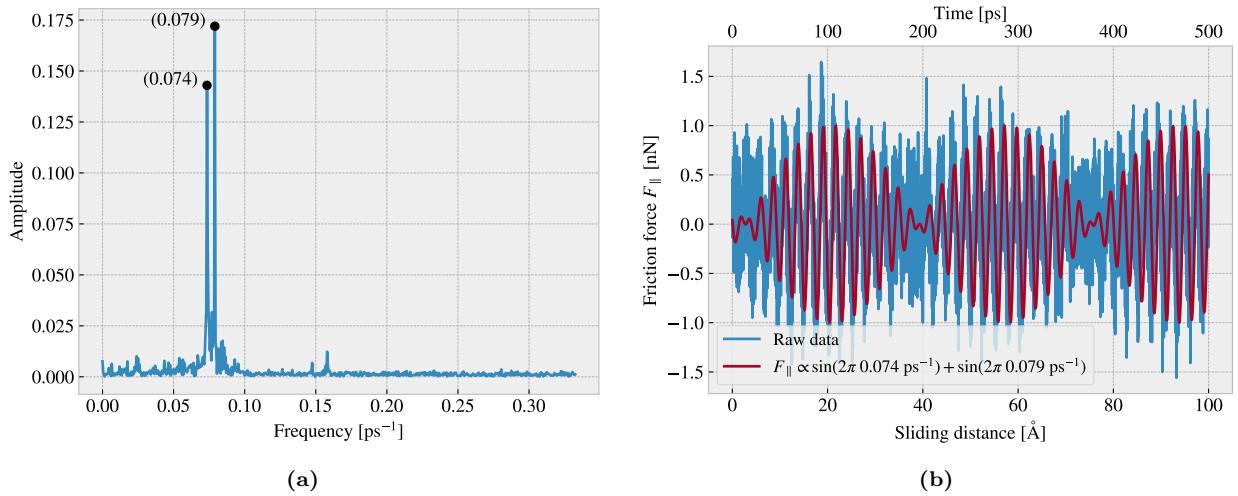


Figure 6.2: Fourier transform analysis of the default parameter force traces shown in Fig. 6.1a and Fig. 6.1b, but with the use of the full data length corresponding to 400 \AA of sliding. (a) The Fourier transform results showing the two most dominant frequency peaks for a reduced frequency range. We excluded higher frequencies from the plot since no significant peaks were found there. (b) A comparison between the wavefunction corresponding to the two frequency peaks found in panel (a) and the raw data force trace from Fig. 6.1b.

⁷The uncertainty Δy is calculated as $\Delta y = \left| \frac{\partial y}{\partial x} \Delta x \right|$ for uncertainty Δx and $y(x)$.

6.2.2 Decompositions

In the previous analysis, we considered the friction force acting on the entire sheet, including the rigid pull blocks, and with respect to the sliding direction. We found this way of measuring the friction force to be the most intuitive and reliable, but we will present the underlying arguments for this choice in the following.

Since we are only applying cuts to the inner sheet, and not the pull blocks, it might appear more natural to only consider friction on the inner sheet. If the desired frictional properties can be achieved by altering the inner sheet one can argue that any opposing effects from the pull blocks can be mitigated by simply adjusting the size ratio between the inner sheet and the pull blocks. However, when looking at the force traces decomposed with respect to the inner sheet and pull block regions respectively in Fig. 6.3a, we observe that the friction force arising from those parts is seemingly antisymmetric. That is, the frictional force exerted by the substrate on the sheet oscillates between the inner sheet and the pull blocks. Keeping in mind that normal force is only applied to the pull blocks we might take this as an intrinsic feature of the system that does not necessarily disappear with a scaling of the size ratio. Any interesting friction properties might depend on this internal distribution of forces. Hence, we hedge our bets and use the full sheet friction force as a holistic approach to avoid excluding relevant features in the measurement data.

Similarly, we might question the decision of only considering the frictional force projected onto the sliding direction as we are then neglecting the “side shift” induced during sliding. In Fig. 6.3b we show the decomposition in terms of the force components parallel F_{\parallel} and perpendicular F_{\perp} to the sliding direction respectively. We notice that the most dominant trend appears for the parallel component. One way to include the perpendicular component is to evaluate friction as the length of the force vector instead. However, this would remove the sign of the force direction and shift the mean friction force up as we see both negative and positive contributions in the parallel force trace. One option to accommodate this issue is by using the vector length for the magnitude but keeping the sign from the parallel component. However, we omit such compromises as this might make the measurement interpretation unnessecary complex, and we use only the parallel component going forward.

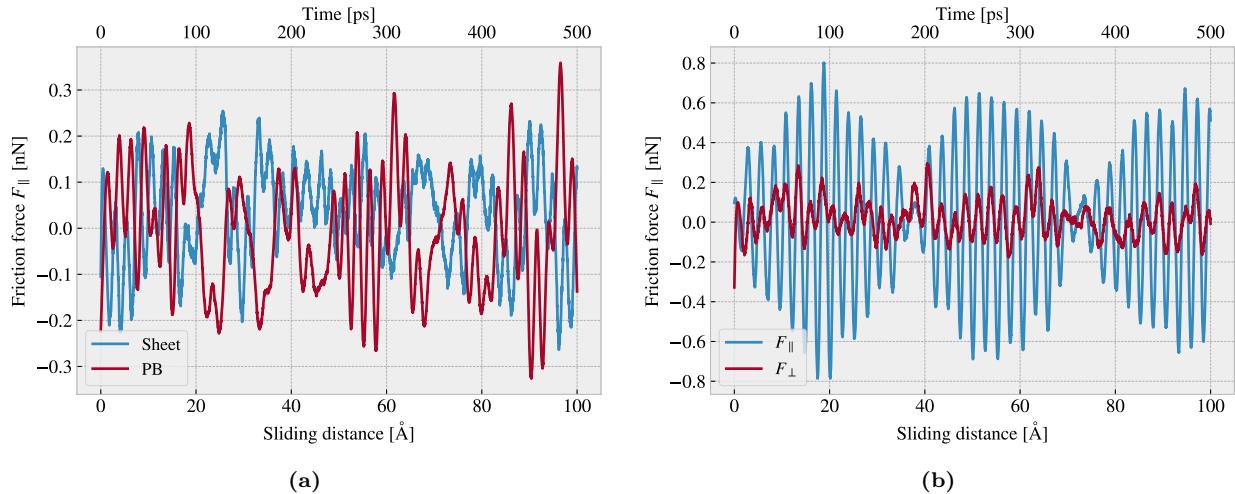


Figure 6.3: Friction force decomposition for the default parameter force trace shown in Fig. 6.1b. We show only the Savgol filters here. (a) Decomposition into the sheet regions inner sheet (sheet) and pull blocks (PB). (b) Decomposition into friction force parallel (F_{\parallel}) and perpendicular (F_{\perp}) to the sliding direction.

6.2.3 Center of mass path

From the previous observations of the force traces in Fig. 6.1 we found both smooth sliding and stick-slip behavior depending on the sliding speed and spring constant. Considering the force decomposition in Fig. 6.3b we know that a frictional force in the perpendicular direction to sliding is also present. By looking at the x, y -position for the sheet Center of Mass (CM) we find a qualitatively different behavior when reconsidering the spring constants and sliding speeds investigated in Fig. 6.1. These results are shown in Fig. 6.4. The default case in Fig. 6.4a shows a rather straight path with only a small side motion in comparison to the cases in Fig. 6.4b and Fig. 6.4c. However, the CM accelerates and deaccelerates with a high frequency, much too high to be associated with the lattice spacing on the order of 2.46 \AA . One possible explanation is that the sheet and substrate constitute an incommensurable contact for which traveling kink excitations make the atoms move in such a way that the sheet CM is incremented in small “bursts”. When looking at the

$K = 10 \frac{\text{N}}{\text{m}}$, $v = 10 \frac{\text{m}}{\text{s}}$ case in Fig. 6.4b we see a completely different CM path where the rapid movements align visually better with the force oscillations shown earlier in Fig. 6.1d. The CM accelerates forward and then deaccelerates in combination with a side motion that leads to the CM path making a loop as it slows down. Finally we have the $K = 10 \frac{\text{N}}{\text{m}}$, $v = 1 \frac{\text{m}}{\text{s}}$ in Fig. 6.4c which is confirmed to have stick-slip behavior from Fig. 6.1f. Here the CM path shows a more chaotic movement between accelerations, but with the rapid parts aligning visually well with the timing of the slips seen in Fig. 6.1f. We might associate the chaotic motion with thermal contributions as these are thought to be dominant at lower sliding speeds.

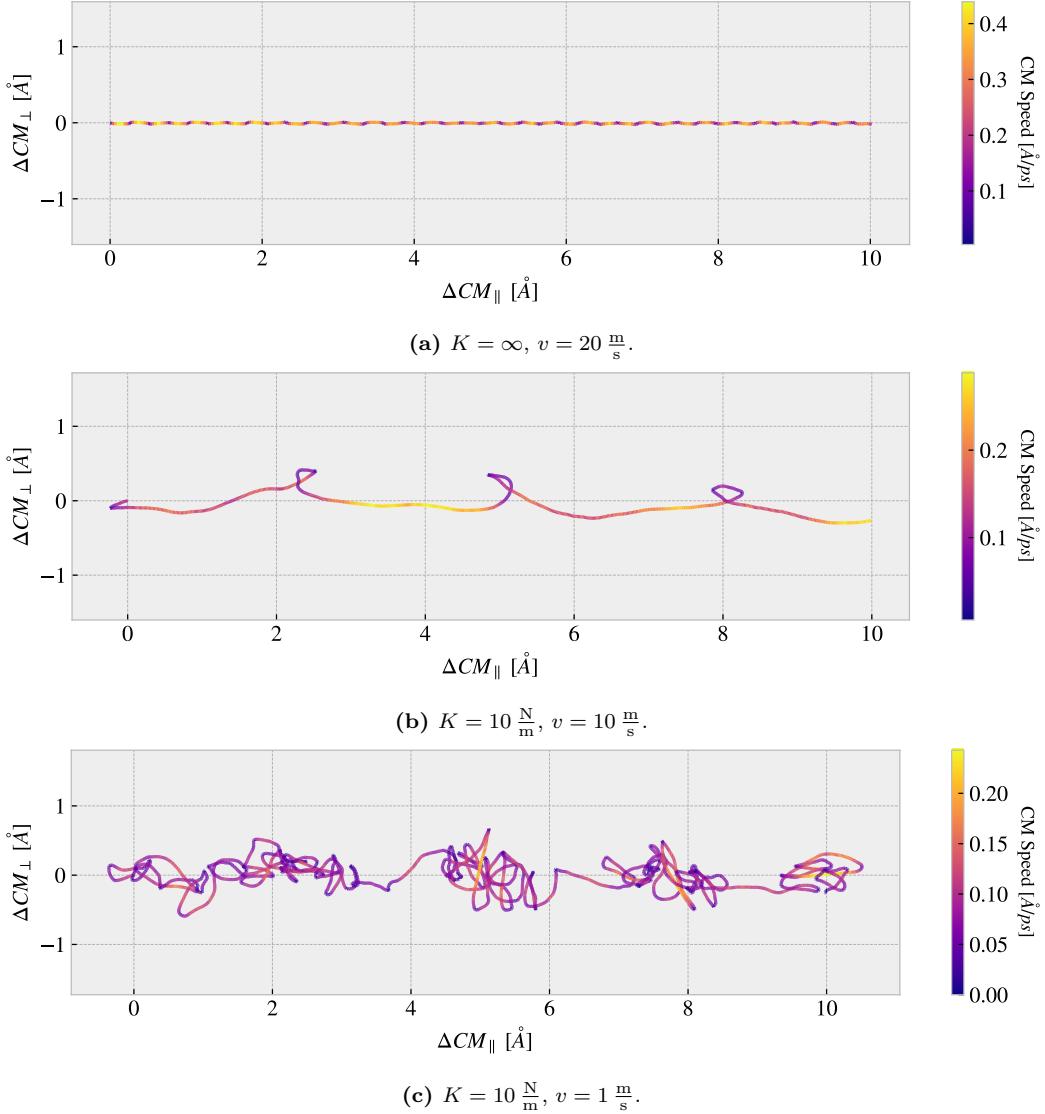


Figure 6.4: Sheet Center of Mass (CM) position relative to the start of the sliding phase in terms of the axis parallel to the sliding direction ΔCOM_{\parallel} and the axis perpendicular to the sliding direction ΔCOM_{\perp} . The colors denote the absolute speed of the CM motion. Panel (a) to (c) shows different parameters used for the spring constant K and sliding speed v similar to that used in Fig. 6.1. (a) Default: $K = \infty$, $v = 20 \frac{\text{m}}{\text{s}}$. (b) Values adopted from Zhu and Li [85]: $K = 10 \frac{\text{N}}{\text{m}}$, $v = 10 \frac{\text{m}}{\text{s}}$. (c) $K = 10 \frac{\text{N}}{\text{m}}$, $v = 1 \frac{\text{m}}{\text{s}}$.

6.3 Defining metrics for friction

In order to evaluate the frictional properties of the sheet we aim to reduce the force trace results, addressed in Sec. 6.2, into single metrics describing the kinetic and static friction respectively.

6.3.1 Kinetic friction

We measure kinetic friction as the mean of the friction force trace. More precisely, we take the mean value of the last half of the dataset in order to ensure that we are sampling from a stable system. For a full sliding

simulation of 400 Å our mean value will be founded on the last 200 Å (1000 ps) of sliding. In Fig. 6.5a we have shown the force trace for the first 10 Å of sliding together with a 50% running mean window. The choice of such a short sliding distance is merely to illustrate the sampling procedure, and we see that the final mean estimate (marked with a dot) takes a negative value due to the specific cut-off of the few oscillations captured here. Nonetheless, one approach to quantifying the uncertainty of the final mean estimate is to consider the variation of the running mean preceding the final mean value. The more the running mean fluctuates the more uncertainty associated with the final estimate. Only the running mean “close” to the ending should be considered, since the first part will rely on data from the beginning of the simulation. From the Fourier analysis in Sec. 6.2.1 we found the longest significant oscillation period to be ~ 71 Å. Hence, we find it reasonable to use the standard deviation (std) for the last ~ 71 Å of the running mean window to evaluate the fluctuations. When including the full sliding length this corresponds to the last $\sim 35\%$ of the running mean window ($400 \text{ \AA} \cdot 50\% \cdot 35\% \approx 71 \text{ \AA}$). We consider the std as an estimate of the absolute error and calculate the relative error by a division of the final mean value. In Fig. 6.5b we showcase a running relative error based on the std, with a window of length 35% of the mean window, in a continuation of the illustrative case from Fig. 6.5a. In this case, we get an extremely high relative error of $\sim 257\%$, but this is consistent with the fact that the short sampling period leads to an unphysical negative value which should be associated with high uncertainty.

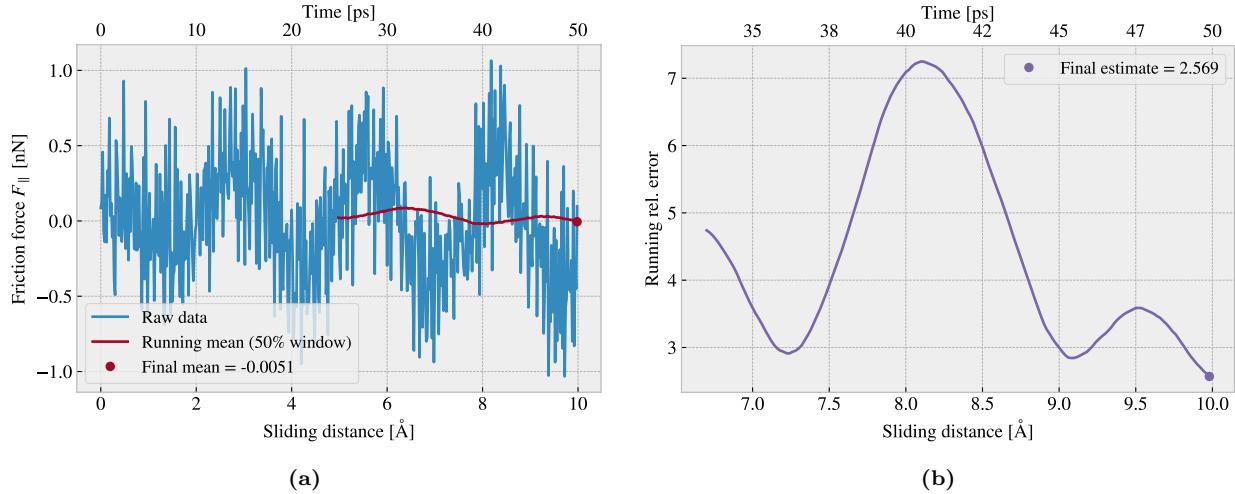


Figure 6.5: Supporting figures for the description of the kinetic friction metric and corresponding uncertainty for an example using a reduced sliding distance of 10 Å. (a) The running mean for the force trace with a window length of 50% of the sliding length in this example. (b) The running relative error calculated using the standard deviation for the 35% running window of the running mean (red line in panel (a)). For both figures, the running value is displayed at the end of their respective windows.

When including the full dataset of 400 Å of sliding, such that the std window actually matches with the longest period of oscillations expected, we get a final relative error of $\sim 12\%$ as shown in Fig. 6.6. This is arguable just at the limit of an acceptable error, but as we shall see later on in Sec. 6.6 this high relative error is mainly associated with the cases of low friction. When investigating different configurations under variation of load and strain we see a considerably lower relative error as the mean friction evaluates to higher values. One interpretation of this finding is that the oscillations in the running mean are not substantially influenced by the magnitude of the friction. In that case, the relative error will spike for the low friction cases, and the absolute error might be the more reliable measure, i.e. using simply the std without dividing by the final mean value.

6.3.2 Static friction

The maximum value is one of the common choices for addressing static friction, even though the definition of static friction is a bit vague. When considering the force traces in Fig. 6.1 we observe that the force oscillations increase in magnitude toward a global peak at ~ 20 Å. Thus, one could be inclined to identify this peak as the maximum value associated with the static friction force. However, as we have already clarified, this steady increase in friction is part of a slower oscillation that repeats with a period of ~ 71 Å. By plotting the top three maximum values recorded during a full 400 Å simulation, for 30 logarithmically spaced load values in the range $[0.1, 100]$ nN, we observe that the global maximum rarely falls within this first oscillation period as shown in Fig. 6.7a. Only 2 out of 30 global maxima and 4 out of 90 top three maxima can be



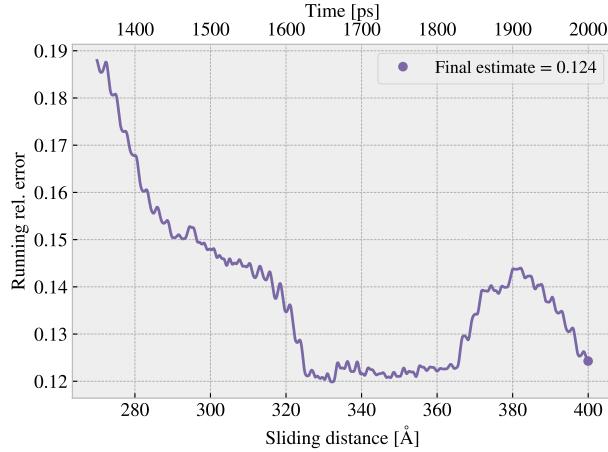


Figure 6.6: Running relative error calculated from the standard deviation (std) for a full 400 Å sliding simulation. The running std window is 70 Å (35% of the running mean window, which is 50% of the full sliding length).

associated with the start of the sliding by this definition. Thus, this result suggests that our default system does not yield a static friction response in the sense of an initial increase in friction due to a depinning of the sheet from the static state. Several modifications to the system parameters may enhance the likelihood of observing a substantial static friction response. These include prolonging the initial relaxation time, as static friction is believed to increase logarithmically with time [28], or increasing the sliding force more slowly by decreasing the sliding speed or utilizing a soft spring tethering. As an attempt to test the latter part of this hypothesis, we conducted a series of simulations with different spring constants, $K \in [5, 200]$ nN, including also $K = \infty$, while maintaining the relaxation period and sliding velocity at their default values. The results shown in 6.7b do not support this hypothesis since the reduction of the spring constant did not lead to the maximum friction peak appearing within the first oscillation period. We acknowledge that this outcome might be compromised by our choice of the relaxation period or sliding speed. However, given the ambiguity surrounding the determination of static friction, we will mainly concern ourselves with kinetic friction in the remainder of this thesis.

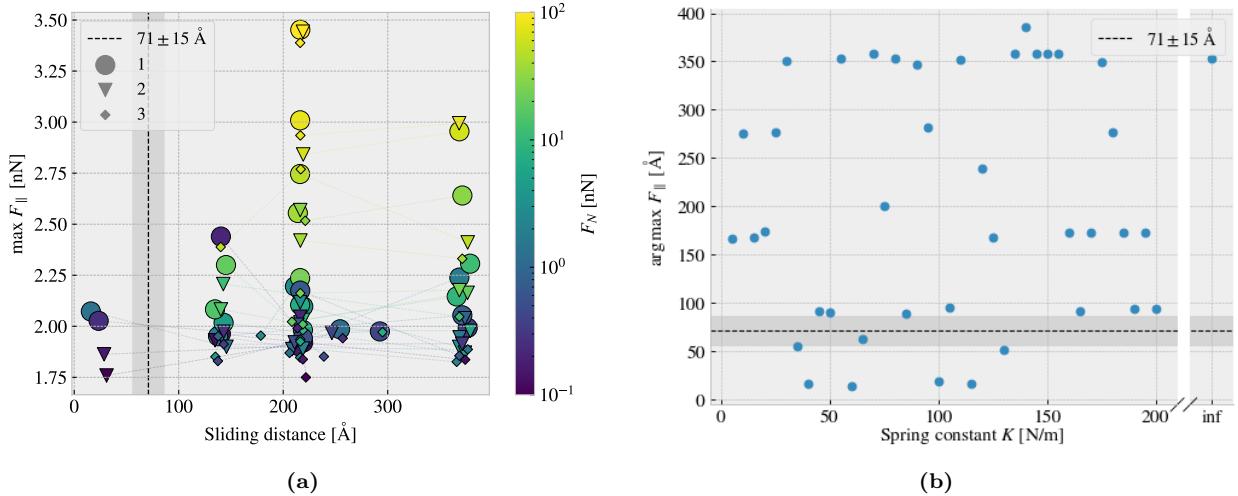


Figure 6.7: Investigation of the sliding displacement corresponding to the maximum friction force peaks. The dotted line, along with the gray-shaded area indicating the degree of uncertainty, represents the slowest significant oscillation period identified in the data from the Fourier analysis in Fig. 6.2a. This line serves as a threshold for determining whether a peak falls within the initial portion of the sliding simulation. (a) The top three maxima for simulations with varying loads; 30 logarithmically spaced loads in the range [0.1, 100] nN. The marker shapes denote the top 1, 2 and 3 respectively and the color denotes the normal load. (b) The sliding displacement corresponding to the friction maxima for simulations with varying spring constant; 40 uniformly spaced values in the range $K \in [5, 200]$ N/m in addition to $K = \infty$.

6.4 Out-of-plane buckling

The out-of-plane buckling is one of the motivations for investigating the application of Kirigami cuts in the context of friction properties. Therefore, we perform a stretch simulation, at low temperature ($T = 5\text{ K}$) without any substrate, in order to verify that we can reproduce an out-of-plane buckling with the Tetrahedron and Honeycomb patterns. For this investigation, we consider the Tetrahedron (7, 5, 1) and the Honeycomb (2, 2, 1, 5) pattern in comparison to the non-cut sheet. We quantify the out-of-plane buckling by assessing the distribution of atoms along the z-direction (perpendicular to the plane) during straining. We calculate the minimum and maximum z-value as well as the atom count quartiles 1%, 10%, 25%, 50% (median), 75%, 90% and 99% as shown in Fig. 6.8. The Tetrahedron and Honeycomb patterns show significant buckling in comparison to the non-cut sheet, which only exhibits minor buckling of $\sim 2\text{ \AA}$, which is of the same order as the lattice spacing. We notice that the Tetrahedron pattern buckles more in consideration of the minimum and maximum peaks while the remaining quantiles seem to be more closely spaced than for the Honeycomb. By addressing the simulation results visually, using the *Open Visualization Tool OVITO*, we find that this can be attributed to fringes on the edge “flapping around” and thus increasing the minimum and maximum values. This is also evident from the simulation with the substrate seen in Fig. A.1j in Appendix A.

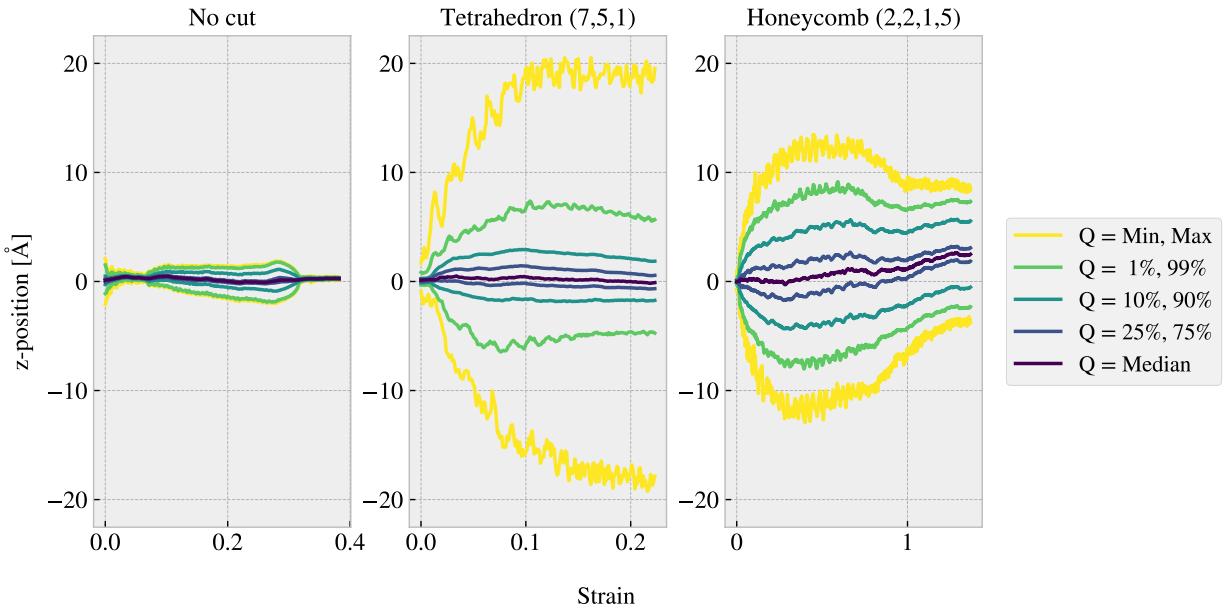


Figure 6.8: The out-of-plane buckling during straining for the non-cut (No cut), Tetrahedron (7, 5, 1), and Honeycomb (2, 2, 1, 5) pattern respectively in vacuum at low temperature $T = 5\text{ K}$. The buckling is quantified by the distribution of the atom z-positions, which are perpendicular to the sheet plane, and the colors indicate selected quantiles. The rupture strain for each pattern, from left to right, is 0.38, 0.22, and 1.37, respectively. The results indicate that the Tetrahedron and Honeycomb patterns exhibit significant out-of-plane buckling in comparison to the non-cut sheet.

Given the confirmation of out-of-plane buckling in a vacuum, as seen in Fig. 6.8, we reintroduce the substrate in order to investigate whether this effect carries over to a changing contact area. For this simulation, we raise the temperature to the default value of $T = 300\text{ K}$. We keep the normal force off and let the sheet stick purely by the adhesion forces between the sheet and substrate. We quantify the contact area through the relative number of atoms in the sheet within chemical range of the substrate. The cut-off for this interaction is set to 4 \AA , adopted from [79], corresponding to $\sim 120\%$ the LJ equilibrium distance. Usually, the contact area is calculated as the number of contacting atoms multiplied by an associated area for each atom. However, since we are not interested in the absolute value of the area, but rather the relative change, we omit the multiplication factor. That is, we consider the relative number of atoms within the contact range as our metric of choice, which is proportional to the contact area. The relative contact for the three configurations No cut, Tetrahedron (7, 5, 1) and Honeycomb (2, 2, 1, 5) during straining are shown in Fig. 6.9. The figure reveals a significant drop in relative contact as the sheets are strained, which agrees qualitatively with the buckling observed in Fig. 6.8 without the substrate. The Honeycomb pattern turned out to be both the most stretchable, with a rupture strain of 1.27, and the one with the biggest decrease in relative contact with a minimum of approximately 43%. Notice, that the relative contact is never actual 1.0 but instead



reaches a maximum of 96% at zero strain. This is attributed to the temperature fluctuations and the choice of cut-off.

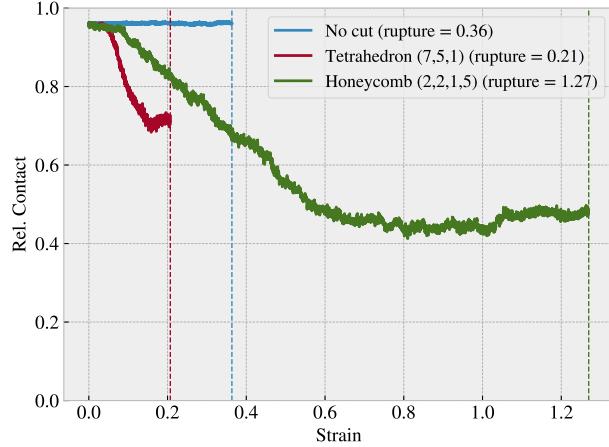


Figure 6.9: The relative contact during straining of the non-cut (No cut), Tetrahedron (7,5,1), and Honeycomb (2,2,1,5) pattern respectively in contact with a substrate. The relative contact is given as the relative number of atoms in the sheet within chemical interaction range. The cut-off for the interaction range is 4 Å corresponding to $\sim 120\%$ the LJ equilibrium distance. No normal force is applied and the temperature is kept at $T = 300$ K.

Selected frames from the simulation result are shown in Appendix A which reveals a bit more information on how the buckling occurs. The Tetrahedron pattern deforms rather quickly and smoothly into small tetrahedron spikes, as the name suggests. In the Honeycomb pattern, on the other hand, the deformations initiate from one side first. As the sheet stretches, more rows of the pattern are activated, producing the honeycomb-looking shape when seen from above. This can also be seen from the flip book animation in the right corner of the pages. Both patterns exhibit a small increase in relative contact when they are approaching their rupture strain, which agrees with the results from Fig. 6.8 where the buckling reduces slightly towards the rupture strain.

6.5 Investigating default parameters

We carry out a more extensive investigation of the friction dependence on temperature T , sliding speed v , spring constant K , and timestep dt . This is done partly to understand how the dependencies relate to the theoretical, numerical and experimental results, and partly to understand how these parameters affect the stability of our system. We use the default parameters presented in Table 6.1 and investigate the results as we change parameters, one at a time. We keep the load at 1 nN. We consider the mean friction force parallel to the sliding direction for the full sheet, sampled from the last half of the simulation as described in Sec. 6.3, representing the kinetic friction. The results are shown in Fig. 6.10, where the shaded area (connected linearly) denotes the absolute error defined by the std as described in Sec. 6.3 as well.

From the temperature investigation in Fig. 6.10a we find an overall increasing kinetic friction with temperature for both the non-cut sheet and the Honeycomb pattern. The Tetrahedron pattern shows both decreasing and increasing trends. The general trend shows a convex curve in the range (30–480 K) with a minimum around our default choice of 300 K, but with rapid fluctuations at the start (10–30 K) and end region (480–500 K). Similar fluctuations are also seen for the Honeycomb pattern, although it shows an underlying increasing trend throughout. When comparing the non-cut sheet and the Honeycomb pattern we observe that the slope for the increasing trend is higher for the Honeycomb pattern. From the predictions of the Prandtl–Tomlinson model and experimental results we expect to find a decrease in friction with increasing temperature which disagrees with our results. However, in similar MD-based studies by Guerra et al. [95] they report that the friction dependence on temperature reverses at high sliding speed. They attribute this to the crossover from the diffusive to the ballistic regime taking place at a sliding speed of 1 to 10 m/s. This agrees with our observations since we are using a sliding speed of 20 m/s. The increase in friction with temperature also agrees with the predictions from the 2D Frenkel-Kontorova model [63]. In the absence of any clear suggestions from the results regarding an appropriate temperature, we take the common choice of using the room temperature 300 K. The non-cut and Tetrahedron friction shows to be rather stable

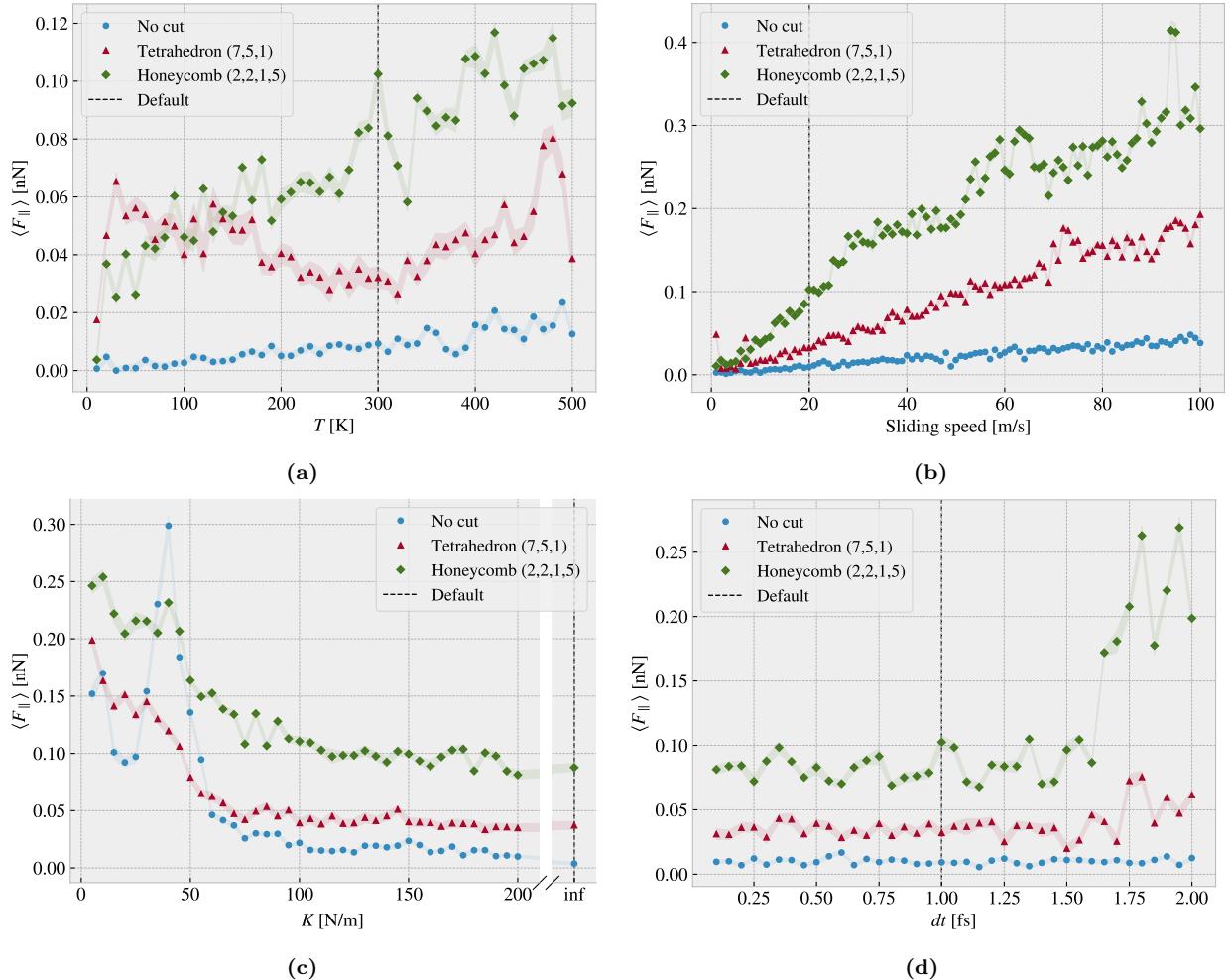


Figure 6.10: Investigation of the friction dependencies to the selected parameters: Temperature T (a) sliding speed (b), spring constant K (c) and timestep dt (d), for the non-cut sheet, the Tetrahedron (7,5,1), and the Honeycomb (2,2,1,5) patterns. The dotted line denotes the default parameter choice and the shaded area denotes the estimated absolute error.

around this temperature choice, although we do see some significant fluctuations for the Honeycomb pattern in this range. However, we do not regard this as a critical feature.

From the sliding speed investigation in Fig. 6.10b we generally find increasing friction with sliding speed. Due to the relatively high sliding speed used in these simulations, we expect a viscous friction $F_{\text{fric}} \propto v$ as reported by Guerra et al. [95] and predicted by the Prandtl–Tomlinson model. In general, this aligns rather well with our results, especially for the non-cut sheet. However, the Tetrahedron and Honeycomb sheets seem to fall slightly into a sublinear relationship as it approaches higher sliding speeds. This behavior could potentially be attributed to velocity saturation, a phenomenon predicted by the Prandtl-Tomlinson model without considering the damping effect of the thermostat. On the other hand, experimental results generally show a logarithmic increase in friction with sliding speed, which may align better with the behavior observed for the Tetrahedron pattern. Furthermore, we find that the Tetrahedron and Honeycomb patterns both display indications of local fluctuations that could be attributed to phonon resonance effects, as discussed in relation to the Frenkel-Kontorova models. Our choice of a sliding speed at 20 m/s mainly reflects a consideration of computational cost, but the fact that no immediate resonance fluctuations appear in the proximity of this value supports the choice further.

From the investigation of the spring constant parameter in Fig. 6.10c, we observe a significant decrease in friction with increasing spring constant. This can be attributed to the transition from a stick-slip influenced regime to a smooth sliding regime as evidenced by the force traces in Fig. 6.1. For soft springs (low K) the friction response is quite sensitive to the specific choice of spring constant which is especially seen for the non-cut sheet around $K = 40$ N/m. Thus, in order to avoid this domain we settled for the infinitely stiff spring ($K = \infty$). This is also considered a more favorable option due to its ability to provide greater standardization of the simulations since it allows for a more controlled movement of the sheet.



Finally, we consider the numerical stability of the simulation result as we vary the simulation timestep in Fig. 6.10d. The results indicate that the simulation is generally stable below a timestep of ~ 1.5 fs, beyond which instability arises for the cut sheets (Tetrahedron and Honeycomb). This confirms that our choice of timestep is reasonable, but we do observe some fluctuations that are more pronounced for the cut sheets. These fluctuations imply that randomness plays a role in our simulations and that the cut sheets exhibit relatively higher instability compared to the non-cut sheet. Further investigation through varying the random seed for the initial velocity and thermostat could shed more light on this matter. In the meantime, we may consider these fluctuations as a sign that the uncertainty in our results is higher than our estimation from using the running mean and running std. For the Honeycomb sheet, these fluctuations are on the order of ± 0.017 nN. This also aligns with the observation that the friction values obtained for the default parameters are not exactly equal across all simulation results in Fig. 6.10. We find the estimated error by the fluctuations with timestep to match these deviations rather well.

6.5.1 Computational cost

Our simulations are carried out on a CPU cluster made available by the University of Oslo. This allows us to run multiple simulations at once and with each simulation running in parallel on multiple CPU cores as well. When selecting the simulation parameters, we also need to keep in mind the computational cost. Given that the chosen parameters will be applied to multiple simulations, any increase in computational time will be multiplied by the number of intended simulations, which is roughly 10,000. The computational cost is especially dependent on the timestep and the sliding speed as this will affect the number of computations needed for a full simulation. As an extension of the investigation from Fig. 6.10 we report on the computational times associated with temperature, sliding speed, spring constant and timestep. By retrieving the computational time used for the parameter investigation in Fig. 6.10 we get the timings shown in Fig. 6.11. Note that these timings are only based on a single simulation for each parameter as opposed to an average over multiple runs which is necessary for more reliable data.

The computational time is governed by the number of timesteps in the simulation and the time used per simulation timestep. For a fixed sliding distance, the number of timesteps in the simulation is inversely proportional to sliding speed and similar inversely proportional to timestep dt . From the results in Fig. 6.11 we find that the sliding speed obeys this expectation rather well by $t \propto v^{-0.977 \pm 0.005}$. The timing did not increase as strongly as expected with timestep length, falling above the $1/dt$ relation with $t \propto dt^{-0.87 \pm 0.02}$. Moreover, we find that increasing temperature also makes for an increased computation time. This can be attributed to an increase in the computation time associated with the force calculations. The rising temperature is associated with more fluctuations in the system which might yield more atoms within the force calculation cutoffs for each computation. This kind of consideration can also be attributed to the reason for the deviating timing relationship with dt . Finally, for the spring constant, we did not see any noticeable effect on timing.

In general, we have selected our simulation parameters: temperature, sliding speed, spring constant, and timestep, based on numerical stability and computational cost. For the timestep, we found that a value of 1 fs, commonly used in similar studies [84, 85], produced stable results while higher values were prone to instabilities and lower values were computationally expensive. The sliding speed was chosen primarily based on computational cost, with a default value of 20 m/s being a reasonable compromise between computational efficiency and relatability to the lower values more commonly used in other studies. Although a lower sliding speed could lead to an interesting regime governed by stick-slip motion, it represents a factor of 20 increase in computational time. Since stick-slip motion is out of reach based on the chosen sliding speed, we found that using an infinitely stiff spring $K = \infty$ was the most reasonable option to ensure stable results. Finally, the temperature investigation did not provide much guidance for a specific choice, so we settled for the standard choice of room temperature $T = 300$ K.

6.6 Strain and load dependencies

So far, we have carried out a general analysis of the system behavior under the influence of various simulation parameters. This lays the foundation for the remaining study as we now shift our intention towards friction dependence on strain and load.

6.6.1 Pressure reference for normal load

We consider a load range of 0.1–10 nN which coincides with the general range investigated in other MD studies [79, 85]. In order to relate the magnitude of this load we provide a short calculation of the

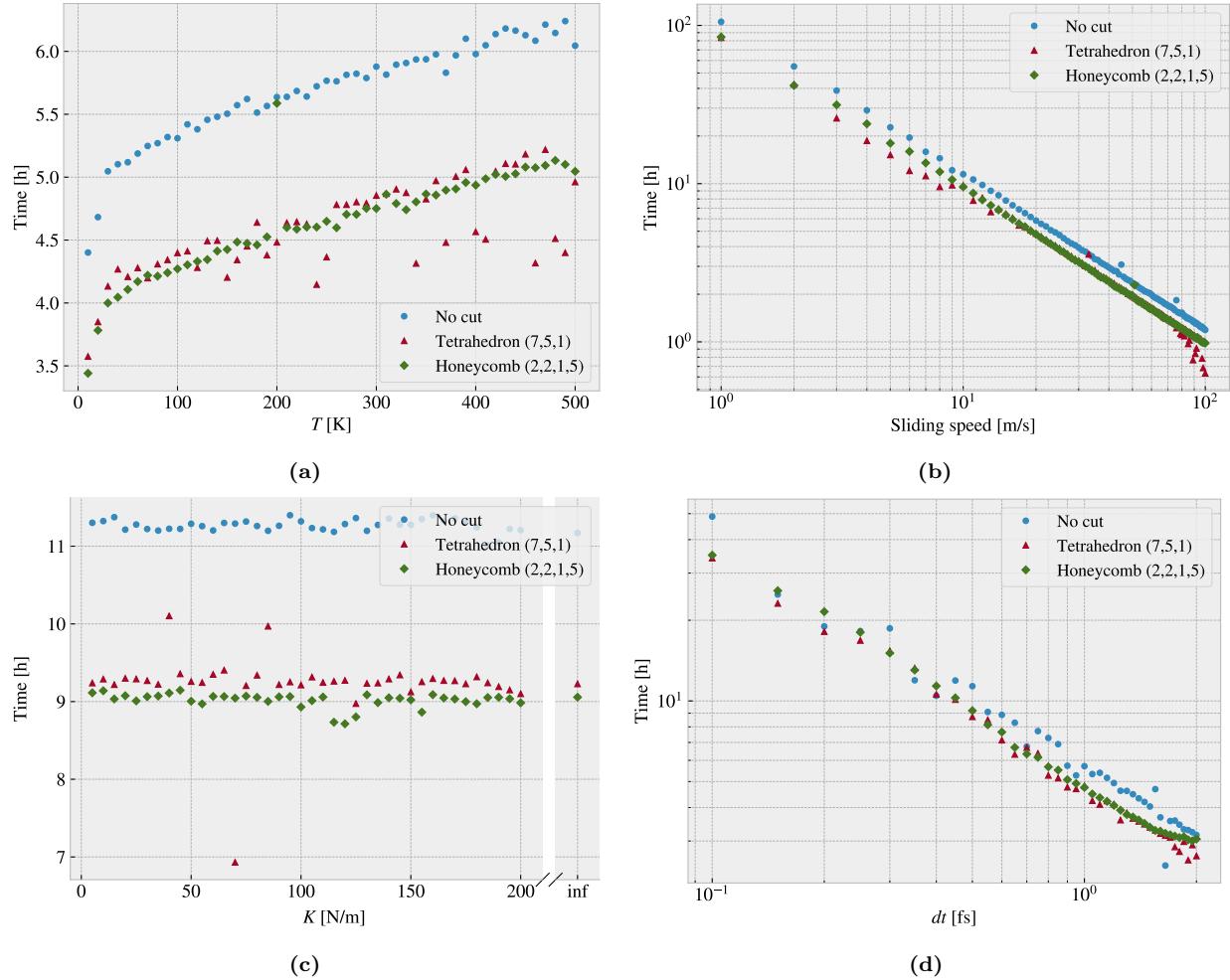


Figure 6.11: The computational timings for the simulations carried out for the parameter investigation in Fig. 6.10 in terms of CPU hours running on 16 cores on the CPU cluster. The timing t with respect to the sliding speed v is found to follow $t \propto v^{-0.977 \pm 0.005}$ while the timing for the timestep length dt follows $t \propto dt^{-0.87 \pm 0.02}$.

corresponding pressure. We will use the pressure underneath a stiletto-heeled shoe as a high-pressure reference from a macroscale perspective. The diameter of a stiletto-heeled shoe can be less than 1 cm [138], and hence an 80 kg man⁸ standing on one stiletto heel, with all the weight on the heel, will generate a pressure

$$P = \frac{F}{A} = \frac{mg}{r^2\pi} = \frac{80 \text{ kg} \cdot 9.8 \frac{\text{m}}{\text{s}^2}}{\left(\frac{10^{-2} \text{ m}}{2}\right)^2 \pi} = 9.98 \text{ MPa.}$$

The fact that the pressure under a stiletto heel can get this high, actually greater than the pressure under an elephant foot, is an interesting realization in itself that is often used in introductory physics courses [139]. Nonetheless, this serves as a reasonable upper bound for human executed pressure. With a full sheet area of $\sim 212 \text{ nm}^2$ our load range of 0.1–10 nN corresponds to a pressure of 0.47–47 MPa which relates reasonably to our macroscale reference. This pressure might be incompatible with various industrial purposes, but with no specific application in mind, this serves as a decent reference point. Notice, that if we consider a human foot with the area 113 cm^2 [140] the pressure drops to a mere 70 kPa corresponding to only ~ 0.01 nN.

6.6.2 Strain dependency

We consider the effects of stretching the sheet using the non-cut, Tetrahedron (7,5,1) and Honeycomb (2,2,1,5) sheet as used so far. While we consider the actions of “stretching” and “straining” to be synonyms we will describe the sheet deformation state with the technical term *strain*. The strain describes the change

⁸Yes, a man can certainly wear stiletto heels.



in length relative to a reference length. We use the relaxed sheet as our reference length L_0 for which the straining towards a larger length L corresponds to the strain $\epsilon = (L - L_0)/L_0$. Thus a strain of 0 corresponds to the relaxed sheet and a strain of 1 corresponds to a doubling of sheet length. Notice that the measure of strain is dimension specific, but we will only consider the y -direction which corresponds to the direction of applied strain in the simulations. For each configuration, we run a rupture test where the given sheet is stretched under zero load, but still under the influence of adhesion from the substrate. The rupture strain is then recorded, and multiple new simulations are initiated with strain values uniformly spaced between zero and the rupture strain. For the load we use 0.1, 1 and 10 nN. First, we aim to reproduce the contact investigation from Fig. 6.9. We quantify the relative contact as described in Sec. 6.4, but we convert this into a single metric for a given simulation by considering the mean of the last 50% of the data points, similar to what we have done for the mean friction. We also adopt a similar method for quantifying the error (see Sec. 6.3). Fig. 6.12a shows a significant decrease in relative contact with strain for the kirigami patterns. This agrees qualitatively with the non-loaded continuous strain simulations in Fig. 6.9. This result implies that the change in contact is not governed by a momentum effect during stretching, as each simulation now keeps a constant strain throughout. From an asperity theory point of view, the reduction in contact area is theorized to induce a linearly dependent reduction in friction. However, when considering the kinetic friction in Fig. 6.12b we find that this is not the case. For the Tetrahedron and Honeycomb patterns, we find that the friction initially increases with a decreasing contact area. Yet, these are not simply inversely proportional as the friction force suddenly dips down and up again, in the strain range of 0.08–0.11 for the Tetrahedron and 0.73–1.05 for the Honeycomb pattern. When considering the non-cut sheet, we find that both the contact area and friction remain seemingly unaffected by strain. This indicates that the contact area is not a dominating mechanism for friction in our system, although the non-linear friction-strain curve might be correlated with a decrease in contact area through an underlying mechanism related to the out-of-plane buckling. This can be supported by the fact that the contact-strain and friction-strain curves for the Honeycomb pattern both show signs of a discontinuous jump around a strain of 0.32.

Since the non-cut sheet exhibits a flat friction-strain profile we cannot attribute the behavior to an increased tension either. Zhang et al. [77] reported that the friction for a graphene sheet decreases with tension which qualitatively contradicts our observation for the Tetrahedron and Honeycomb patterns and supports that tension is not a governing mechanism for these observations. Instead, we might attribute the results to a commensurability effect which is predicted by the Frenkel-Kontorova models and observed in both numerical and experimental studies. We might also point to a change in the structural stiffness which is reported as an important feature by H. J. Kim and D. E. Kim [37].

We notice that the initial friction value of the non-strained sheet varies among different configurations, with the non-cut sheet exhibiting the lowest initial friction value ($\sim 7 \times 10^{-4}$ nN), the Tetrahedron sheet having a slightly higher value (~ 0.03 nN), and the Honeycomb sheet exhibiting the highest (~ 0.07 nN). This is more clearly shown in the parameter investigation in Fig. 6.10. This supports the idea of a commensurability effect as the removal of atoms might affect the lattice structure of the sheet slightly. However, the magnitude of this effect is generally one order of magnitude lower than the strain-induced friction effect observed for the Tetrahedron and Honeycomb patterns. We attribute this to the idea that the reorganization of the contacting atoms has significantly more impact on commensurability than removing some atoms from the lattice. We notice also that the two orders of magnitude increase in normal load did not make a significant difference in the results. The estimated absolute error marked by a shaded area in Fig. 6.12b were fairly low for both friction on the order of 10^{-3} nN and relative contact on the order of 10^{-4} .

By considering the increase in friction from zero strain towards the first peak of the friction-strain curve we find that the Tetrahedron pattern exhibits a relative friction increase of ~ 27.7 while the Honeycomb pattern exhibits a relative increase of ~ 22.4 . This is in itself an interesting result, but considering that the friction drops almost as dramatically afterward is even more unexpected. For the Tetrahedron pattern, the friction drops by ~ 0.51 nN during an increased strain $\Delta\epsilon \sim 0.04$ while for the Honeycomb pattern the friction drops ~ 0.98 nN during a strain increase of $\Delta\epsilon \sim 0.36$. These results are promising for the aim of achieving a negative friction coefficient for a system with coupled load and strain. We will discuss this further at the end of this chapter in Sec. 6.6.4.

6.6.3 Load dependency

From the investigation of the strain dependency we saw that increasing the normal load from 0.1 to 10 nN did not make a considerable impact on the friction in comparison to the effect associated with strain. One special feature of our system is that we only apply load to the pull blocks, and thus we might suspect this to be of importance. Therefore, we investigate the friction under varying loads for a non-cut sheet comparing the case of loading only the pull blocks against a uniform loading of the entire sheet as shown in Fig. 6.13.

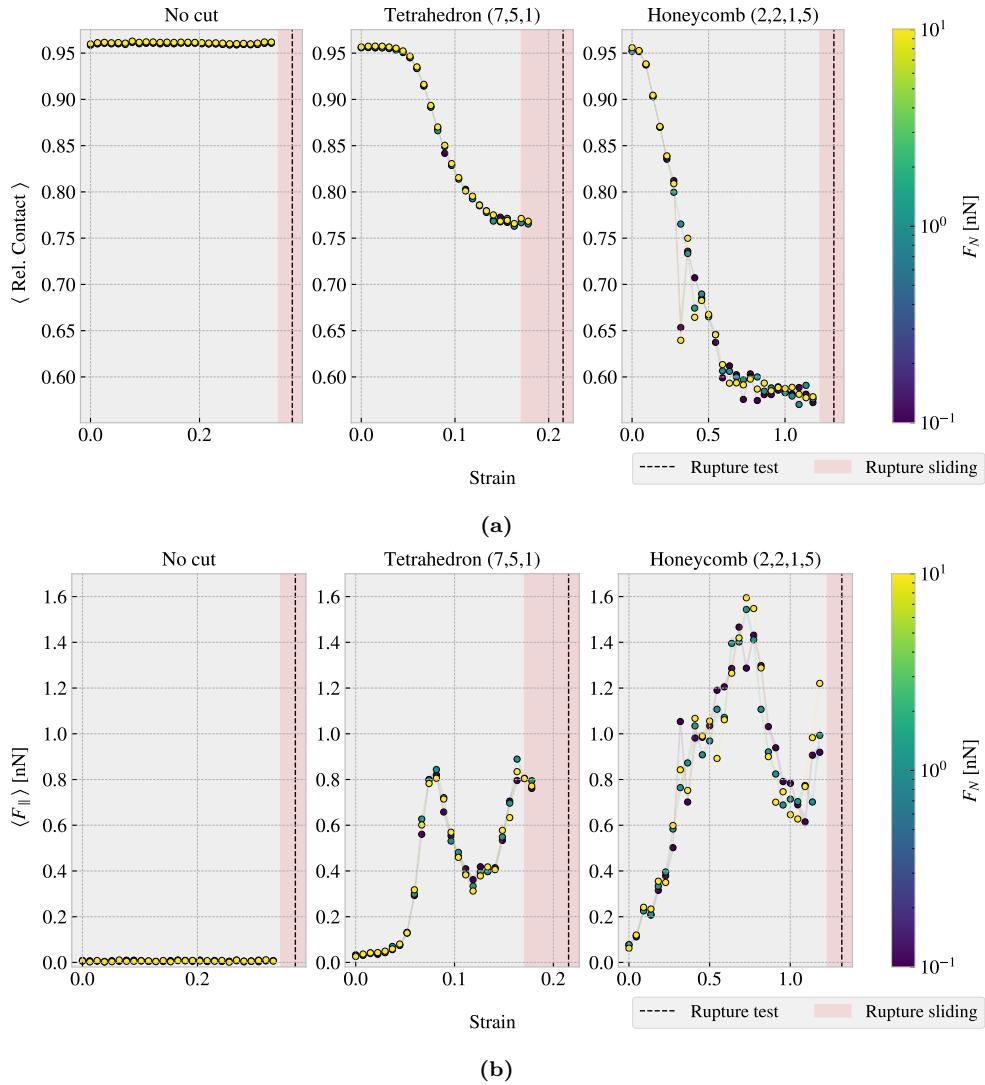


Figure 6.12: Investigation of the frictional behavior for the non-cut, Tetrahedron (7, 5, 1) and Honeycomb (2, 2, 1, 5) sheet under varying strain, consisting of 30 strain values uniformly spaced between 0 and the rupture strain in combination with loads 0.1, 1 and 10 nN. The shaded area, connecting the dots linearly, denotes the the absolute error. The red shade denotes the strain range where ruptures occurred during loaded sliding while the black-dotted line represents the rupture point in the non-loaded rupture test. (a) The average relative contact defined as the relative number of atoms within a contact threshold of 4 Å to the substrate. The absolute error is generally on the order 10^{-4} . (b) The average mean friction force. The absolute error is generally on the order 10^{-3} nN.

Both load distributions show a seemingly non-dependent relationship between friction and load considering the size of the estimated errors. Thus, we do not find any indications that the uniform loading changes the qualitative behavior of our system.

To examine the relationship between friction and normal load as the Kirigami patterns undergo strain, we conduct additional simulations with logarithmically spaced normal load values in the range of 0.1 to 100 nN. This is done for a selected subset of strain stages from Fig. 6.12, with 30 load values for each strain. We also monitor the relative contact during the increased load. The results are shown in Fig. 6.14. From Fig. 6.14b we observe that the three orders of magnitude load increase yields a small increase in friction for all Kirigami patterns. This effect is more pronounced in the figure for the non-cut sheet, but this is mainly because the friction axis shows a narrower range for this figure. It should be noted that due to the logarithmic scale of the load axis, any apparent linear trends in the figure are in fact sublinear. Nonetheless, as the normal load approaches 100 nN, there is some indication of an increase in friction reminiscent of a linear relationship. However, it is difficult to draw firm conclusions as the magnitude of the increase in friction is relatively small compared to the level of noise in the data.



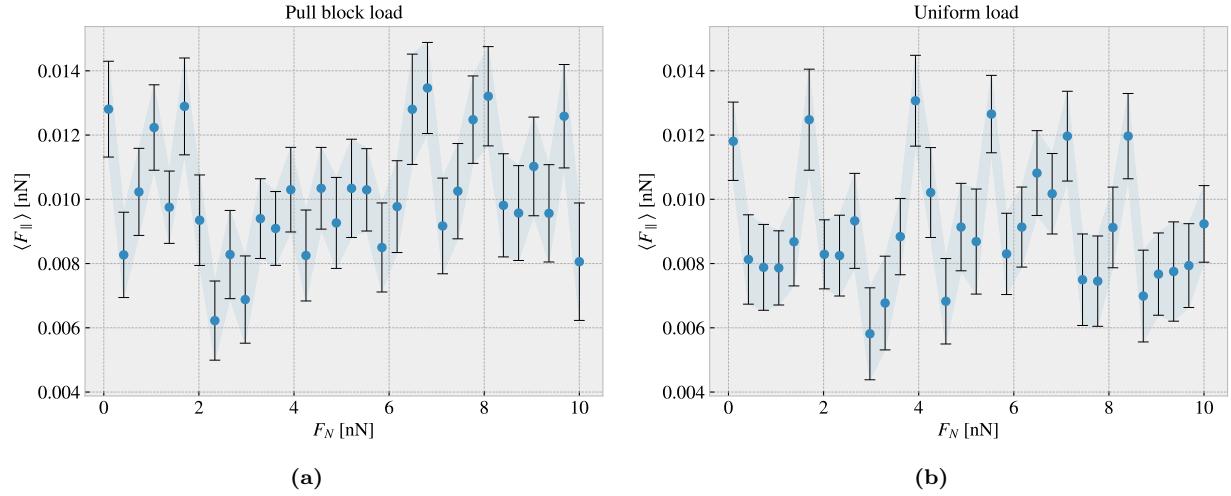


Figure 6.13: Multiple simulations of a non-cut sheet under various normal load values uniformly spaced in the range $F_N \in [0.1, 10]$ nN for two different variations of loading distribution. The shading and the error bars denote the absolute error. (a) The normal loading is applied to the pull blocks only. (b) The normal loading is applied uniformly to the full sheet.

For the investigation of contact area in Fig. 6.14a, we generally find an increasing contact area with load. By considering the full load from 0.1 nN to 100 nN we find the largest increase in relative contact, among the various values of relative strain, to be 0.007 for the non-cut, 0.014 for the Tetrahedron and 0.025 for the Honeycomb sheet. This constitutes a rather small change and thus it is difficult to assess how this relates to a dependency between the contact area and the friction. However, we do see that the Tetrahedron and Honeycomb patterns exhibit a larger change in contact area which aligns with the idea that the out-of-plane buckling makes them behave more like an asperity system. When plotting the friction against relative contact in Fig. 6.14c we do not find any significant evidence for a linear relationship between contact area and friction as otherwise proposed by asperity theory. Instead, from the Tetrahedron and Honeycomb pattern, we confirm that the strain-induced effects are dominant in comparison to any effects from the contact area. By studying an even larger load range we might get more insight into this relationship. Note that we have omitted the error bars in Fig. 6.14 for visual purposes, but the absolute errors for both the relative contact and friction force are on the same order of magnitude as shown in both Fig. 6.12 and Fig. 6.13.

From the friction measurements in Fig. 6.14b we find that the non-cut sheet generally produces a friction force in the range 0.005–0.0025 nN throughout the 0.1–100 nN load range. Using a ratio based friction coefficient definition Eq. (2.2a), $\mu_1 = F_{\text{fric}}/F_N$, this would lead to a coefficient roughly in the range

$$\mu_1, \text{ Eq. (2.2a): } \text{No cut} \sim [10^{-4}, 0.13], \quad \text{Tetrahedron} \sim [4 \times 10^{-4}, 8.7], \quad \text{Honeycomb} \sim [9 \times 10^{-4}, 15.2].$$

However, these values mainly reflect the poorness of this definition, as we find the values to diverge at small loads and decrease toward high loads due to the lacking linear relationship and an offset in the load curve corresponding to a finite friction at zero load. This offset is drastically enhanced for the Kirigami patterns under strain. Due to the small changes in friction compared to the noise in the data, it is not sensible to calculate the slope dF_{fric}/dF_N as a function of load. Nonetheless, if we force a linear fit for the whole range and use the second definition Eq. (2.2b) as $\langle \mu_2 \rangle = \Delta F_{\text{fric}}/\Delta F_N$, we get average coefficients in the range

$$\mu_2, \text{ Eq. (2.2b): } \text{No cut} \sim [4, 9] \times 10^{-5}, \quad \text{Tetrahedron} \sim 5 \times [10^{-5}, 10^{-4}], \quad \text{Honeycomb} \sim [1, 9] \times 10^{-4},$$

depending on the strain values. These numbers should be interpreted cautiously, but we can take it as a rough estimate of the friction coefficient being on the order 10^{-4} – 10^{-5} . This relates to the finding by [66] who reported a seemingly non-existing relationship between friction and normal load for a graphene sheet with changes in friction that corresponds to friction coefficients in the range of 10^{-3} – 10^{-4} when using the slope definition Eq. (2.2b). This supports the idea that the graphene sheet exhibits superlubric behavior in these conditions and that the initial non-strained sheet is in an incommensurable phase. Thus any strain-induced effects might introduce a more commensurable phase that aligns with the observed increase in friction and the fact that the friction never gets smaller than the initial friction value. The non-monotonic increase in friction can then be attributed to transitions in and out of commensurable phases. We found that these strain-induced effects on friction are generally separate from the dependence to load. That is, the introduction of strain mainly shifts the friction-load curve but does not significantly alter its slope.

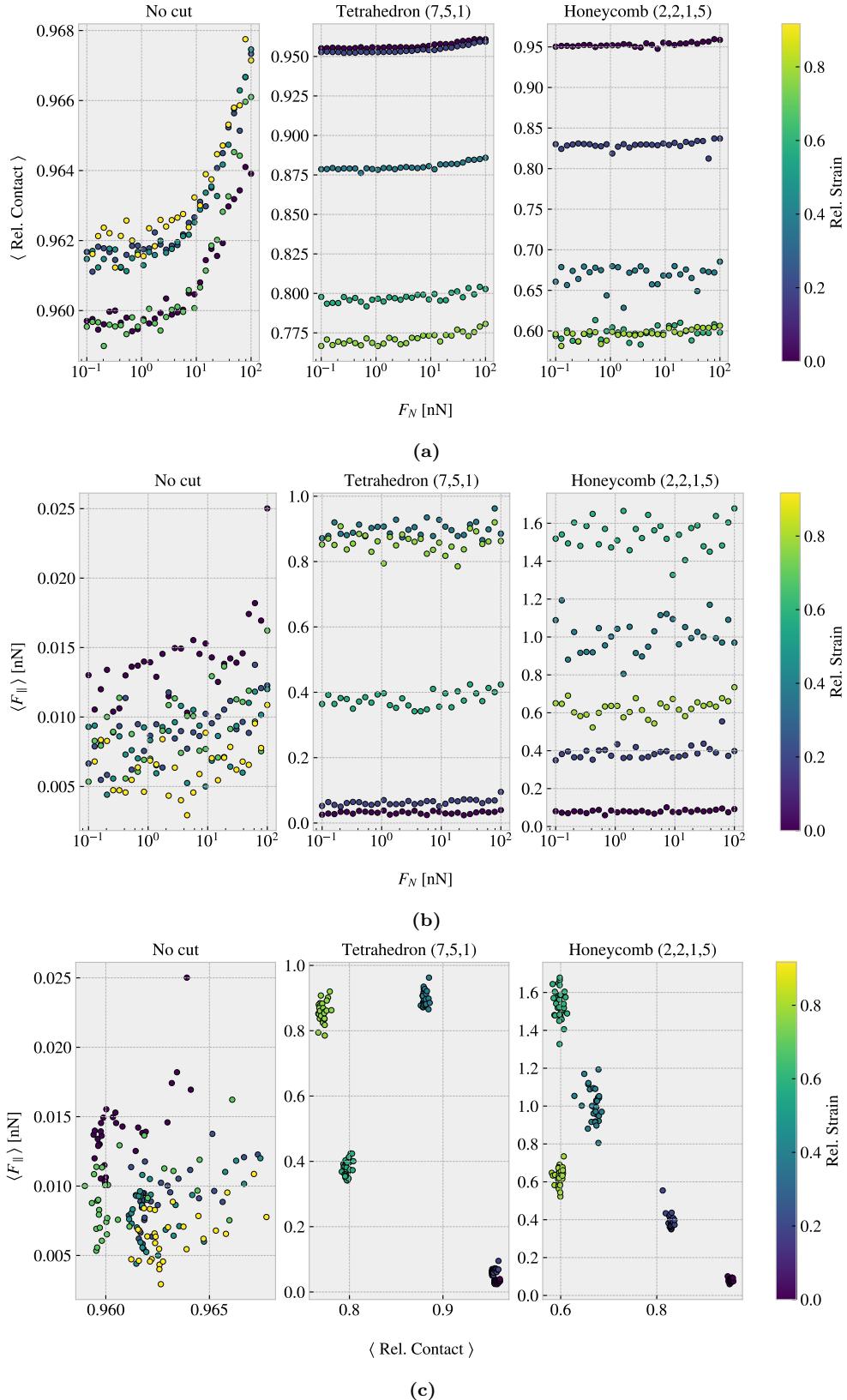


Figure 6.14: Investigation of the frictional behavior for the non-cut, Tetrahedron (7, 5, 1) and Honeycomb (2, 2, 1, 5) sheet under varying load, consisting of 30 logarithmically spaced load values in the range 0.1–100 nN, at different strain stages relative to their rupture strains. We omitted the indication of the absolute errors as they were similarly low as seen in Fig. 6.12. (a) The average relative contact vs. load. (b) The average mean friction vs. load. (c) The average mean friction vs. the average relative contact.



6.6.4 Prospects for achieving a negative friction coefficient

Considering the results from Sec. 6.6.2 and Sec. 6.6.3 we find that strain-induced friction effects, on the order of 1 nN, are generally dominating in comparison to load-induced effects on the order of 0.01 nN given a load range of 0.1–100 nN. This is promising for the idea of achieving a negative friction coefficient for a nanomachine system that couples load and strain. By applying load on the nanomachine we would increase both the load and the strain on the sheet simultaneously. However, since the friction dependency to strain dominates in comparison to load effects such a system can in practice be designed entirely by considering the strain dependency. The friction coefficient is by our definition (Eq. (2.2b)) given as the slope of the friction F_{fric} vs. normal force F_N curve. Hence, for two points $\{(F_{N,1}, F_{\text{fric},1}), (F_{N,2}, F_{\text{fric},2})\}$, $F_{N,1} < F_{N,2}$ we can evaluate the associated friction coefficient $\mu_{1,2}$ as

$$\mu_{1,2} = \frac{F_{\text{fric},2} - F_{\text{fric},1}}{F_{N,2} - F_{N,1}} = \frac{\Delta F_{\text{fric}}}{\Delta F_N}.$$

If we neglect load effects, $F_{\text{fric}}(F_N, \varepsilon) \sim F_{\text{fric}}(\varepsilon)$, and consider a load-strain coupling $\varepsilon = RF_N$ with linear coupling ratio R we get

$$\mu_{1,2}(\varepsilon_1, \varepsilon_2) = \frac{\Delta F_{\text{fric}}(\varepsilon_1, \varepsilon_2)}{\frac{1}{R}(\varepsilon_2 - \varepsilon_1)} = R \frac{\Delta F_{\text{fric}}(\varepsilon_1, \varepsilon_2)}{\Delta \varepsilon}. \quad (6.1)$$

When considering the ratios $\Delta F_{\text{fric}}/\Delta \varepsilon$ for the reduction in friction with strain for the Tetrahedron and Honeycomb patterns in Sec. 6.6.2 we find the corresponding coupled system friction coefficients to be

$$\text{Tetrahedron: } R \frac{-0.51 \text{ nN}}{0.04} = -R \cdot 12.75 \text{ nN}, \quad \text{Honeycomb: } R \frac{-0.98 \text{ nN}}{0.36} = -R \cdot 2.72 \text{ nN}. \quad (6.2)$$

This showcases that we might be able to utilize the strain effect to achieve a negative friction coefficient for the system of coupled load and strain. This hypothesis is further investigated in Chapter 8.

Chapter 7

Kirigami configuration exploration

Building upon the discoveries of the Pilot Study Chapter 6, we will further explore the impact of Kirigami designs on strain-dependent friction. Our focus is primarily to optimize the friction force and friction coefficient toward their maximum or minimum values. To achieve this goal, we will utilize MD simulations to generate an extended dataset that encompasses a wider range of Kirigami designs. This is motivated by the aim of gaining a broader understanding of the friction-strain relationship. We will then leverage this dataset to explore the potential of employing machine learning for the prediction of friction behavior based on Kirigami design, strain, and load. Finally, we utilize the developed machine learning model for an accelerated search for new Kirigami designs.

7.1 Generating the dataset

We create a dataset that contains an extended series of Kirigami design configurations based on the pattern generation methods developed in Chapter 5 for which we will vary the strain and load for each configuration. For each configuration, we sample 15 strain values between 0 and the rupture strain using a pseudo-uniform distribution, meaning that we divide the given interval into equal segments and pick a value from each segment by a uniform distribution. This is due to numerical limitations in LAMMPS⁹, but we find that this gives evenly spaced values which also carry some randomness. Since the normal load did not prove to be dominant in the friction description we only sample 3 normal load values per configuration, uniformly sampled in the range [0.1, 10] nN. In total, this gives $3 \times 15 = 45$ data points for each configuration. For the remaining parameters, we use the default values shown in Table 6.1. We are mainly concerned with the mean friction and whether the sheet ruptures during the simulation. However, we also include the maximum friction, the relative contact, the rupture strain (from the rupture test) and the porosity (void fraction) in the dataset. We generate 68 configurations of the Tetrahedron pattern type, 45 of the Honeycomb type and 100 of the Random walk type. For the Tetrahedron and Honeycomb patterns, we choose a random reference position which results in translational variances of the patterns. A summary of the dataset is given in Table 7.1 while all configurations are shown visually in Appendix B. The Tetrahedron and Honeycomb parameters are chosen to provide additional variations of the configurations evaluated in Chapter 6 which exhibited interesting properties. The Random walk parameters are chosen to introduce as much variety as possible in order to contribute to a wide distribution of configurations in the dataset. Notice that not all submitted data points “make it” to the final dataset, which is due to a small bug in the data generation procedure¹⁰.

⁹In LAMMPS, we sample the various strain values by storing restart files during the straining of the sheet. The restart values are stored at specific timesteps governed by a LAMMPS variable. Such variables allow for a vector of uniform randomly chosen values, but unfortunately, we are not able to sort the vector for ascending values. This will lead to the script waiting to store the restart file according to the next timestep value in the unsorted vector. As soon as the next timestep value is less than the current timestep the program will stop producing restart files and thus skip most of them. However, by first defining a series of intervals we can draw a uniform number for each interval without getting into trouble.

¹⁰The issue arises from the fact that the rupture point in the rupture test does not always match the rupture point in the following simulations. After performing the rupture test the simulation is restarted with a new substrate size, corresponding to the measured rupture strain limit, but also with a new random velocity and thermostat initialization. The sheet is then strained and checkpoints of the simulation state, restart files, are stored for each of the targeted strain samples. However, if the rupture point arrives earlier than suggested by the rupture test, due to randomness from the initialization, some of the planned strain samples do not get a corresponding restart file. Thus, these data points are not included in the dataset even though they ideally should have been noted as a rupture event. This could have been mitigated by a rewrite of the code, but it was first discovered after the dataset had been created. Despite the issue, the dataset still contains a notable 11.57% of rupture events, which is deemed sufficient for the machine learning model to learn to identify ruptures. Therefore we conclude that this issue is not critical for machine learning training.



Table 7.1: Summary of the number of generated data points in the dataset. Due to slight deviations in the rupture strain and the specific numerical procedure not all submitted simulations are included in the final dataset. Notice that the Tetrahedon (7, 5, 2) and Honeycomb (2, 2, 1, 5) from the pilot study are rerun as a part of the Tetrahedon and the Honeycomb datasets separately. However, the reference point for the patterns is randomized and thus these configurations are not fully identical. This is the reason for the ambiguousness in the total sum.

| Type | Configurations | Submitted data points | Final data points | Ruptures |
|-------------|----------------|-----------------------|-------------------|----------------|
| Pilot study | 3 | 270 | 261 | 25 (9.58 %) |
| Tetrahedon | 68 | 3060 | 3015 | 391 (12.97 %) |
| Honeycomb | 45 | 2025 | 1983 | 80 (4.03 %) |
| Random walk | 100 | 4500 | 4401 | 622 (14.13 %) |
| Total | 214 (216) | 9855 | 9660 | 1118 (11.57 %) |

7.2 Data analysis

In order to gain insight into the correlations in the data we calculate the correlation coefficients between all variable combinations. More specifically, we calculate the Pearson product-moment correlation coefficient which is defined, between data set X and Y , as

$$\text{corr}(X, Y) = \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{\langle (X - \mu_X)(Y - \mu_Y) \rangle}{\sigma_X \sigma_Y} \in [-1, 1], \quad (7.1)$$

where $\text{Cov}(X, Y)$ is the covariance, μ the mean value and σ the standard deviation. The correlation coefficients range from a perfect negative correlation (-1) through no correlation (0) to a perfect positive correlation (1). The correlation coefficients are shown in Fig. 7.1. We especially notice that the mean friction force $\langle F_{\parallel} \rangle$ has a significant positive correlation with strain (0.77) and porosity (0.60). However, the relative strain, scaled by the rupture strain, has a weaker correlation of only 0.25. This indicates that the correlation might be associated with the flexibility of the configurations since these can be taken to higher absolute values of strain. This is further supported by the fact that the mean friction and the rupture strain are also strongly positively correlated (0.78). We also observe that the contact is negatively correlated with the mean friction (-0.67) and the strain value (-0.74). This is generally consistent with the trend observed in the pilot study in Fig. 6.12 where the increasing strain was associated with a decreasing contact and mainly increasing mean friction. However, we must note that the correlation coefficient is a measure of the quality of a forced linear fit on the data. Since we have observed a non-linear trend between friction and strain (Fig. 6.12b) we should not expect any near 100% correlations. Additionally, we also notice that all correlations to normal load are rather low, which aligns well with the findings in the pilot study.

Fig. 7.2 shows a visualization of the data (excluding the pilot study configurations) for a subset of variable pairs on the axes. This allows us to visually identify some of the correlations and gain a qualitative understanding of the variations in different planes of the feature space which we eventually will base our machine learning model on.

7.3 Properties of interest

In the pilot study (Chapter 6) we found promising results for the idea of achieving a negative friction coefficient under the assumption of a system with coupled normal load F_N and strain ε . Hence, we will consider this as a main property of interest for our further exploration. We assume that the friction dependence on load is negligible $F(F_N, \varepsilon) \sim F(\varepsilon)$ in comparison to that on strain, and propose a coupling $\varepsilon = RF_N$ with linear coupling ratio R . From these assumptions, we can in practice substitute load for strain in the expression for the friction coefficient of our coupled system $\mu \propto \Delta F_f(\varepsilon)/\Delta \varepsilon$ as shown in Eq. (6.1). This justifies the search for a negative slope on the friction-strain curve since this can be related to a negative friction coefficient in our proposed coupled system.

The remaining question is then how to evaluate the strength of this property. By definition, the minimum (most negative) slope value would give the lowest friction coefficient. However, two data points with a small $\Delta \varepsilon$, corresponding to a small denominator in Eq. (6.1), would potentially lead to a huge negative slope value without any significant decrease in friction. Hence, we choose to consider the decrease in friction with increasing strain as a better metric. Numerically we compute this by locating the local maxima on the friction-strain curve and then evaluating the difference to the succeeding local minima. The biggest difference corresponds to the *max drop* property which serves as our indicator for a negative friction

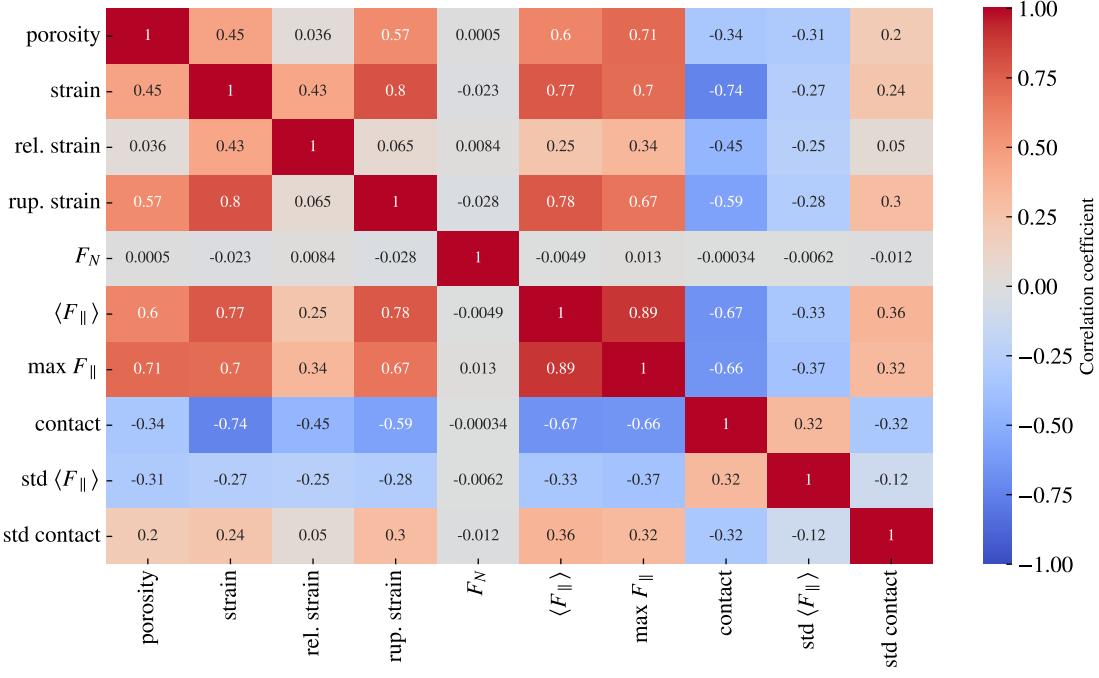


Figure 7.1: Pearson product-moment correlation coefficients Eq. (7.1) for the full dataset (see Table 7.1). Here the relative strain refers to the strain relative to the rupture strain.

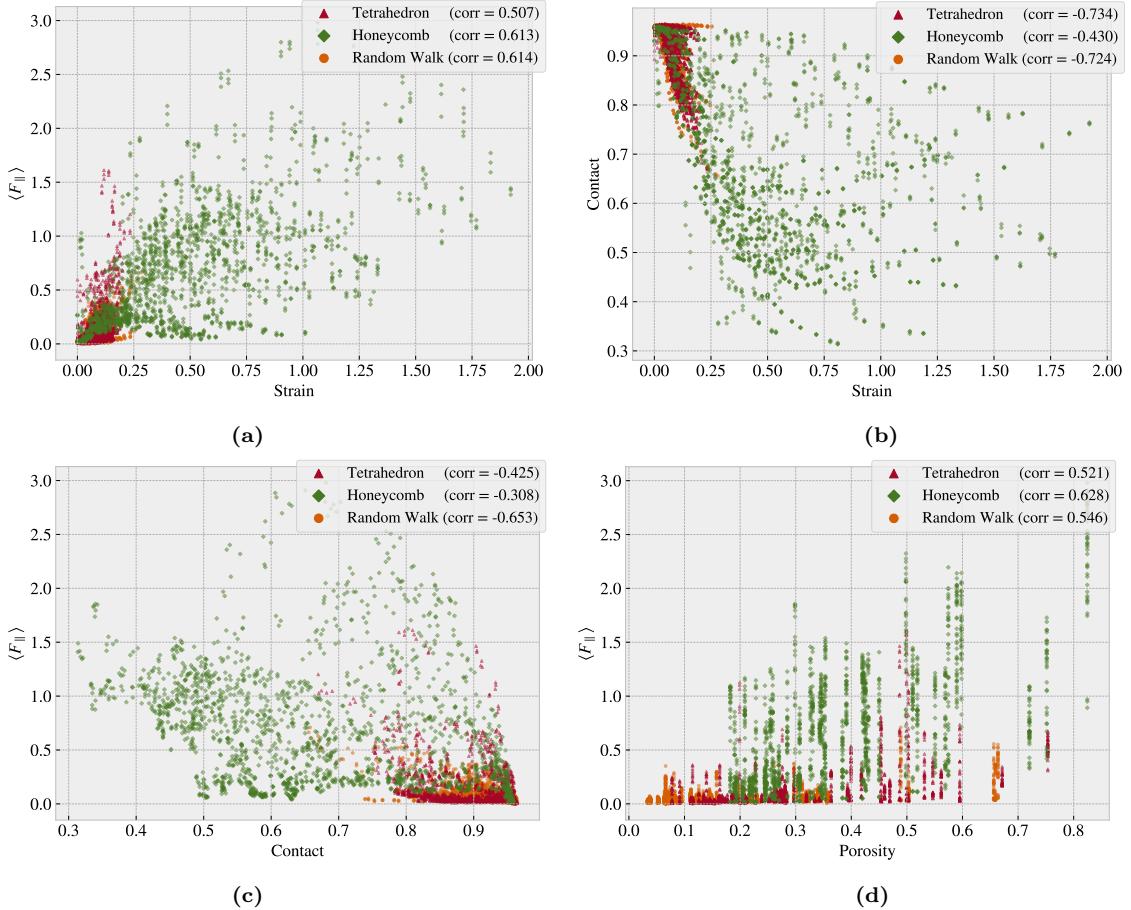


Figure 7.2: Scatter plot of various dataset feature pairs for the Tetrahedron, Honeycomb and Random Walk subsets respectively. The legends include the correlation coefficient in parentheses. (a) Mean friction vs. strain. (b) Relative contact vs. strain. (c) Mean friction vs. relative contact. (d) Mean friction vs. porosity.



coefficient. In this evaluation, we do not guarantee a monotonic decrease of friction in the strain range corresponding to the max drop, but when searching among multiple configurations this is considered a decent strategy to highlight configurations of interest worthy of further investigation. In addition to the max drop property, we also consider the minimum, $\min F_{\text{fric}}$, the maximum, $\max F_{\text{fric}}$ and the maximum difference, $\max \Delta F_{\text{fric}} = \max F_{\text{fric}} - \min F_{\text{fric}}$ for the friction-strain curve. The extrema of these four properties for each of the categories: Tetrahedron, Honeycomb, Random walk and pilot study, are summarized in Table 7.2. The corresponding friction-strain profiles and configurations are shown in Fig. 7.3a to 7.3d (excluding the friction-strain profiles already shown in the pilot study Chapter 6). The friction-strain profiles for the full dataset are shown in Appendix C.

Table 7.2: Evaluation of the properties of interest for the dataset. Each table shows the top scores for each of the four properties within each of the separate data categories: Tetrahedron, Honeycomb, Random walk and pilot study (the three configurations used in the pilot study). The tables denote the names of the top candidate configurations, the relevant strain values for the property and the property values themselves. For the Tetrahedron and Honeycomb category, we compare the top candidate scores to the scores of the Tetrahedron (7, 5, 1) and Honeycomb (2, 2, 1, 5) pattern used in the pilot study in the right-most column.

| Tetrahedron | Configuration | Strain | Value [nN] | Tetrahedon (7, 5, 1) [nN] |
|-------------------------------|---------------|------------------|------------|---------------------------|
| $\min F_{\text{fric}}$ | (3, 9, 4) | 0.0296 | 0.0067 | 0.0262 |
| $\max F_{\text{fric}}$ | (5, 3, 1) | 0.1391 | 1.5875 | 0.8891 |
| $\max \Delta F_{\text{fric}}$ | (5, 3, 1) | [0.0239, 0.1391] | 1.5529 | 0.8603 |
| max drop | (5, 3, 1) | [0.1391, 0.1999] | 0.8841 | 0.5098 |

| Honeycomb | Configuration | Strain | Value [nN] | Hon. (2, 2, 1, 5) [nN] |
|-------------------------------|---------------|------------------|------------|------------------------|
| $\min F_{\text{fric}}$ | (2, 5, 1, 1) | 0.0267 | 0.0177 | 0.0623 |
| $\max F_{\text{fric}}$ | (2, 1, 1, 1) | 1.0654 | 2.8903 | 1.5948 |
| $\max \Delta F_{\text{fric}}$ | (2, 1, 5, 3) | [0.0856, 1.4760] | 2.0234 | 1.5325 |
| max drop | (2, 3, 3, 3) | [0.5410, 1.0100] | 1.2785 | 0.9674 |

| Random walk | Configuration | Strain | Value [nN] |
|-------------------------------|---------------|------------------|------------|
| $\min F_{\text{fric}}$ | 12 | 0.0562 | 0.0024 |
| $\max F_{\text{fric}}$ | 96 | 0.2375 | 0.5758 |
| $\max \Delta F_{\text{fric}}$ | 96 | [0.0364, 0.2375] | 0.5448 |
| max drop | 01 | [0.0592, 0.1127] | 0.1818 |

| Pilot study | Configuration | Strain | Value [nN] |
|-------------------------------|-------------------|------------------|------------|
| $\min F_{\text{fric}}$ | No cut | 0.2552 | 0.0012 |
| $\max F_{\text{fric}}$ | Hon. (2, 2, 1, 5) | 0.7279 | 1.5948 |
| $\max \Delta F_{\text{fric}}$ | Hon. (2, 2, 1, 5) | 0.7279 | 1.5325 |
| max drop | Hon. (2, 2, 1, 5) | [0.7279, 1.0463] | 0.9674 |

From the property comparison in Table 7.2, we find that both the Tetrahedron and Honeycomb subsets contain improved candidates for each of the property scores in comparison to the Tetrahedron (7, 5, 1) and Honeycomb (2, 2, 1, 5) examined in the pilot study. Overall, the Honeycomb pattern type is resulting in the highest scores for the maximum properties while the minimum friction is still achieved by the non-cut sheet. This latter observation confirms the findings of the pilot study since our dataset does not provide any indication that friction can be reduced for a Kirigami sheet under strain. However, the improvement in the remaining properties indicates that the dataset contains valuable information that can provide a direction for further optimization of the maximum properties. Considering the Random walk we find that the max property scores are generally lower than those of the Tetrahedron and Honeycomb patterns. However, since these are found to be on a comparable order of magnitude we argue that these contribute relevant information for the frictional dependency to Kirigami configurations. The Random walk patterns exhibit greater diversity compared to the other patterns and therefore provide some immediate insights into which structures can be associated with each of the properties of interest. For the $\min F_{\text{fric}}$ top candidates (Fig. 7.3a) we find that the Random walk candidate has a rather low cut density (low porosity) and vertical cuts. Since these cuts run parallel to the stretching direction one can hypothesize that this minimizes the induced buckling effect which agrees with the relatively flat contact-strain curve. For the minimum candidate of the Tetrahedron pattern, we also observe a low decrease in contact area, and in both these cases this corresponds with a seemingly flat friction-strain curve as well. When considering the remaining friction-strain curves throughout Fig. 7.3a

to 7.3d we find that a rising friction-strain curve is always seen together with a declining contact-strain curve. This supports the general observation of a correlation between the strain-induced friction effects and the contact area. When looking at the 96th Random walk pattern, which is the top candidate for both the max F_{fric} (Fig. 7.3b) and max Δf_{fric} (Fig. 7.3c) properties, we find a rather porous configuration with mainly horizontal-orientated cuts. This has some structural reminiscence with the general shape of the Honeycomb pattern. Finally, for the Random walk max drop candidate, Random walk pattern 01, we do see a small drop in friction. Although, this is not as significant as seen for the Tetrahedron and Honeycomb candidates. We notice that the configuration contains some slanted cuts which might be reminiscent of parts of the general Tetrahedron pattern.



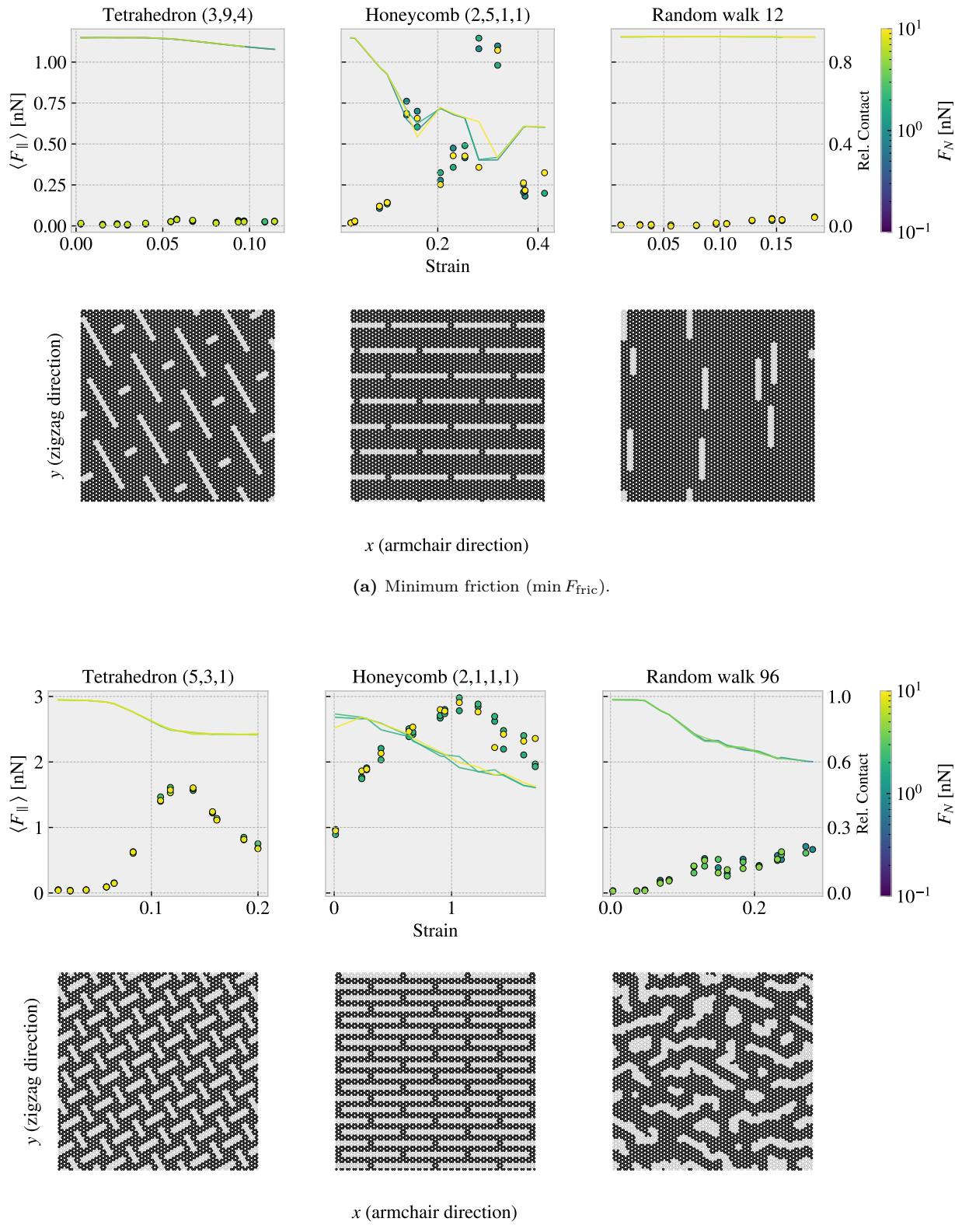


Figure 7.3: (The figure continues on the next page)

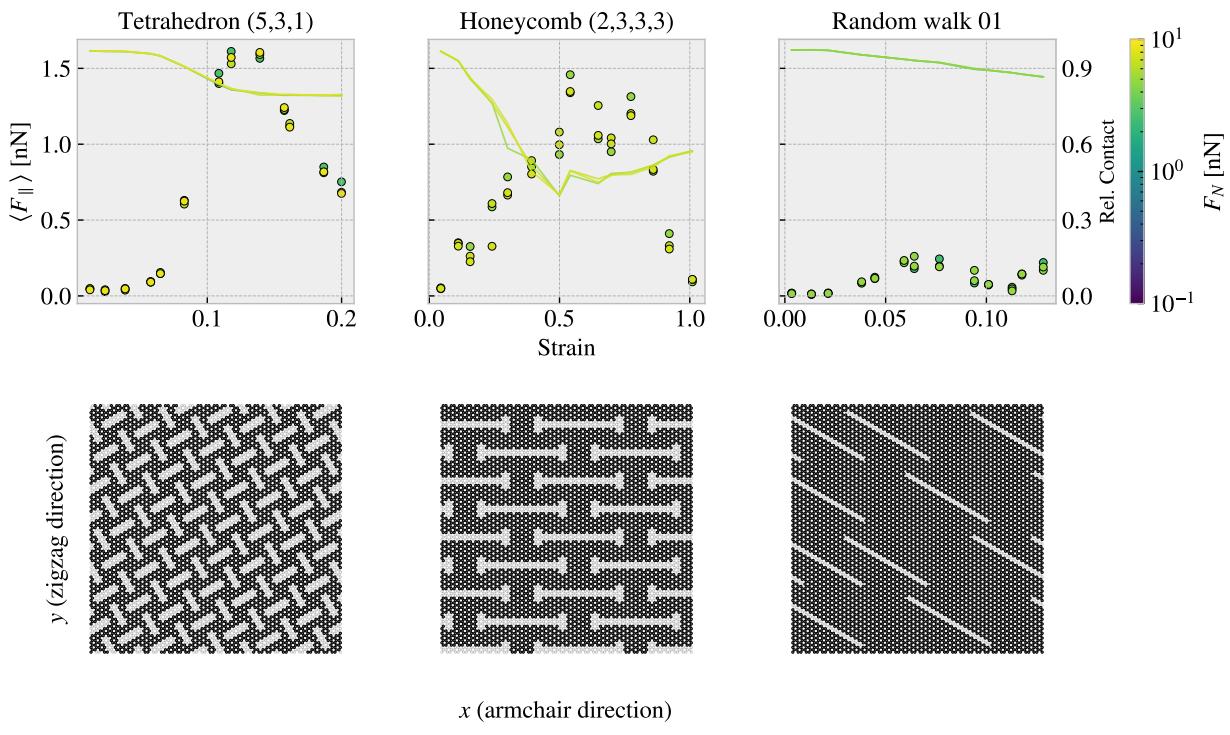
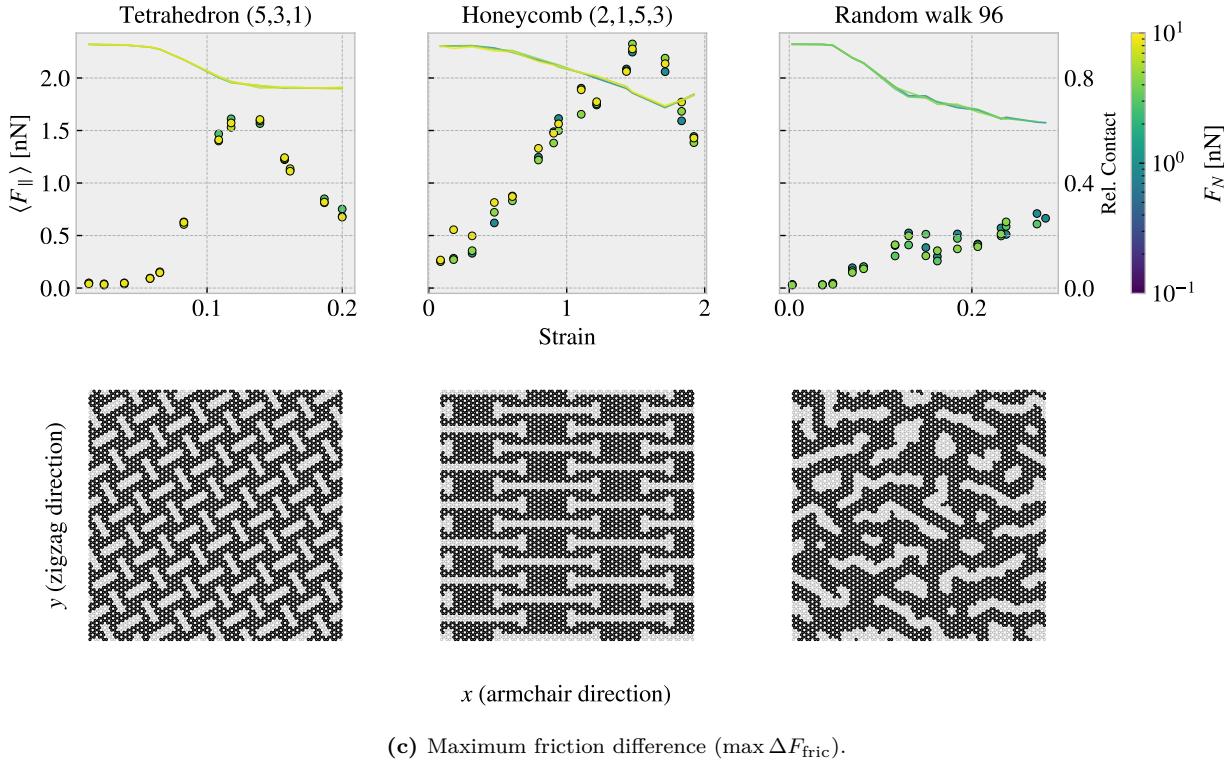


Figure 7.3: Illustration of the top candidates within each of the four properties of interest. The upper row shows the friction-strain curve with dotted points (friction value on the left y-axis) and the contact-strain curve with a solid line (relative contact value on the right y-axis). The bottom row shows the corresponding Kirigami patterns. (a) Minimum friction ($\min F_{\text{fric}}$). (b) Maximum friction ($\max F_{\text{fric}}$). (c) Maximum friction difference ($\max \Delta F_{\text{fric}}$). (d) Maximum drop in friction ($\max \text{drop}$).



7.4 Machine learning

Given the MD-based dataset presented in the previous section, containing Honeycomb, Tetrahedron and Random walk geometries, we investigate the possibilities of training a machine learning model to predict the friction behavior from a given Kirigami configuration, strain and load.

7.4.1 Architecture

Due to the spatial dependencies in the Kirigami configurations, we use a convolutional neural network (CNN). Similar studies which predict mechanical properties for graphene sheets have used a VGGNet style of network, Hanakata et al. [6, 7] and Wan et al. [8], which we adopt for this study as well. The VGGNet-16 architecture illustrated in Fig. 7.4 shows the key features that we will include:

- The image is processed through a series of 3×3 convolutional filters (the smallest size capable of capturing spatial dependencies) using a stride of 1 with an increasing number of channels throughout the network. We use zero padding to conserve the spatial size during a convolution. Each convolutional layer is followed by a ReLU activation function.
- The spatial dimensions are reduced by a max pooling layer, filter size 2×2 and a stride of 2, which halves the spatial resolution each time.
- The latter part of the network consists of a fully connected part using the ReLU activation as well. The transition from the convolutional to the fully connected part is achieved by applying a filter with the same dimensions as the last convolutional feature map. This essentially performs a linear mapping from the spatial output to the fully connected layer where the number of channels corresponds to the nodes in the first fully connected layer.

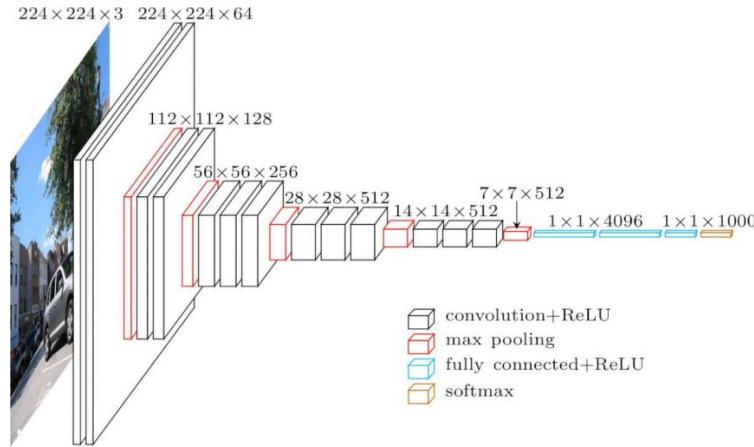


Figure 7.4: Illustration of the convolutional network architecture for the VGGNet-16 proposed by K. Simonyan and A. Zisserman [141]. Reproduced from [142].

We deviate from the VGGNet-16 architecture by including batch normalization and restricting ourselves to setting up the convolutional part of the network in terms of the convolutional block (Convolution \rightarrow Batch normalization \rightarrow ReLU \rightarrow Max pooling). Similarly, we define a fully connected block by two elements (Fully connected \rightarrow ReLU) which match the VGGNet model. Hanakata et al. and Wan et al. used a similar architecture with the parameters

$$\begin{array}{ll} \text{Hanakata et al. [6]} & C16 \ C32 \ C64 \ D64, \\ \text{Wan et al. [8]} & C16 \ C32 \ D32 \ D16, \end{array}$$

where C denotes a convolutional block with the number denoting the number of channels, and D a fully connected (dense) block with the number denoting the number of nodes. For the purpose of determining a suiting complexity for the architecture, we adopt the approach by Wan et al. [8] who used a “staircase” pattern for combining the convolutional and fully connected blocks. By defining a starting number of channels S for the first convolutional layer and a network depth D we fill the first half of the network layers with convolutional blocks, doubling in channel number for each layer, and the latter half with fully connected

blocks halving the number of nodes in a reverse pattern. For instance, the architecture *S4D8* will take the form

$$\text{Input} \rightarrow \underbrace{\text{C4} \text{ C8} \text{ C16} \text{ C32} \text{ D32} \text{ D16} \text{ D8} \text{ D4}}_{D=8} \rightarrow \text{Output.} \quad (7.2)$$

This provides a simple description where S and D can be varied systematically for a grid search over architecture complexity.

7.4.2 Data handling

7.4.2.1 Input

We use three variables as input: Kirigami configuration, strain of the sheet and applied normal load. The Kirigami configuration is given as a two-dimensional binary matrix while the strain and load are both scalar values. This gives rise to two different options for the data structure:

1. Expand the scalar values (strain and load) into 2D matrices of the same size as the Kirigami configuration matrix by copying the scalar value to all matrix coordinates. This can then be merged into an image of three channels used as a single input.
2. Pass only the Kirigami configuration through the convolutional part of the network and introduce the remaining scalar values directly into the fully connected part of the network halfway in.

Both options utilize the same data, but the latter option is more directed toward independent processing of the data while the first makes for an intertwined use of the configuration, strain and load input. We implemented both options but found immediately that option 1 was producing the most promising results during short training tests, and thus we settled for this data structure.

7.4.2.2 Output

For the output, we are mainly concerned with mean friction and the rupture detection. In combination, these values will make the model able to produce a friction-strain curve with an estimated stopping point as well. However, in order to retain the option to explore other relations in the data we include the maximum friction, relative contact, porosity and rupture strain in the output as well. Notice that we weigh the importance of these output variables differently in the loss as described in Sec. 7.4.3.

7.4.2.3 Data augmentation

In order to increase the utility of the available data one can introduce data augmentation. For most classification tasks this usually includes distortions such as color shifts, zoom, reflect etc. However, such distortions are only valid since the classification network should still classify a cat as a cat even though it is suddenly a bit brighter or flipped upside down. For our problem, we can only use augmentation that matches a physical symmetry. Such a symmetry exists for reflection across the y-axis, and thus we perform this transformation with a 50% chance for both training and validation data. We cannot use a reflection across the x-axis as the sheet is sliding in a positive y-direction. Such a transformation would correspond to a change in the sliding direction which we cannot expect to be fully symmetric.

7.4.3 Loss

The output contains two different types of variables: scalar values and a binary value (rupture). For the scalar values we use the mean squared error Eq. (4.1) and for the binary output we use binary cross entropy Eq. (4.2). We calculate the total loss as a weighted sum of the loss associated with each output

$$L_{tot} = \sum_o W_o \cdot L_o.$$

We choose the weights W_o to be 1/2 for the mean friction and 1/10 for the remaining 5 output variables, thus sharing the loss evenly for the remaining 50% of the weight. During the introductory phase of the model implementation, we tried different settings for these weights, but we found that the results varied little. Hence, we concluded that this was of minor importance and we settled on the values defined above.



7.4.4 Hypertuning

For the hypertuning we focus on architecture complexity, learning rate, momentum and weight decay. We use the ADAM optimizer with the initial default values of $\beta_1 = 0.9$, $\beta_2 = 0.999$ and zero weight decay, for which we will vary momentum β_1 and weight decay in the hypertuning. We use a batch size of 32 and train the model for a maximum of 1000 epochs while storing the best model based on the validation scores. Since the learning rate is considered to be one of the most important hyperparameters we will determine a suitable choice for the learning rate using the learning rate range test for each of the two grid searches:

1. Architecture complexity grid search of S vs. D with individually chosen learning rates for each complexity combination.
2. Momentum vs. weight decay grid search with learning range chosen with regard to each momentum setting.

We consider first the architecture complexities in the range $S \times D = \{2, 4, 8, 16, 32, 64, 128, 256\} \times \{4, 6, 8, 10, 12, 14\}$. For each architecture complexity, we perform an initial learning rate range test and determine the suitable choice for the learning rate as the point for which the validation loss decreases most rapidly. The learning rate is increased exponentially within the range 10^{-7} to 10 with increments for each training batch iteration. This is done for a single epoch where a batch size of 32 yields a total of 242 possible increments. This corresponds to an exponent increment of approximately $1/30$ giving a relative increase $10^{1/30} \sim 108\%$ per batch iteration. The learning rate range test is presented in Fig. 7.5 for various model complexities. We notice that the suggested learning rate decreases with an increasing number of model parameters. This decrease is further independent of the specific relationship between S and D .

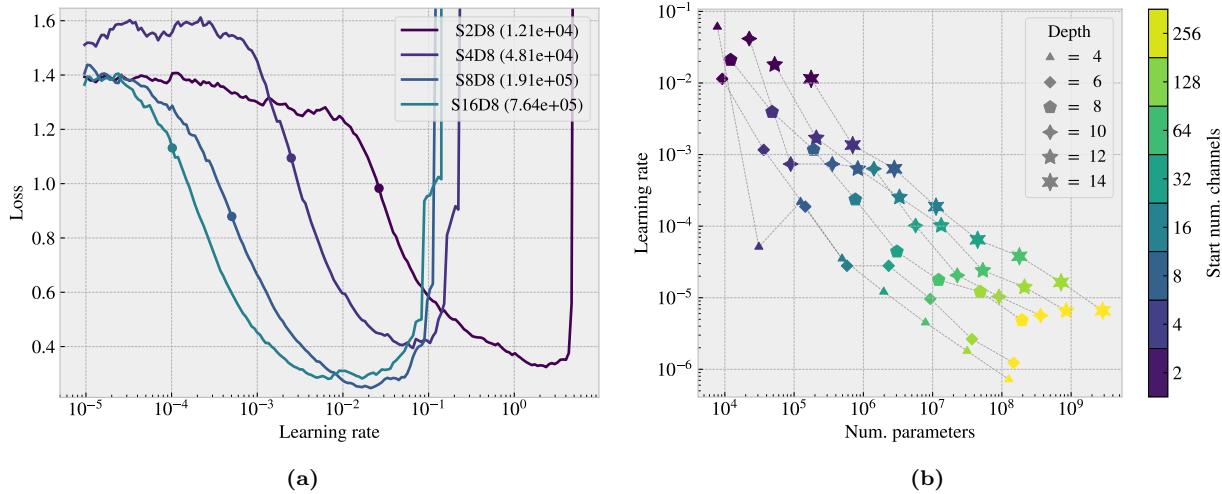


Figure 7.5: Learning rate range test for various model complexities. We increase the learning rate exponentially from 10^{-7} to 10 during one epoch corresponding to an exponent increment of roughly $1/30$ per batch iteration. (a) A few examples of the validation loss as a function of the learning rate. The exemplary architectures are $S[2, 16]D8$ with the corresponding number of model parameters shown in parentheses in the legend. The dots indicate the suggested learning rate at the steepest decline of the validation loss. (b) The full results showing the suggested learning rates depending on the number of model parameters with color coding differentiating the number of start channels S and marker types differentiating different model depths D .

With the use of the suggested learning rates from Fig. 7.5 we perform a grid search over the corresponding S and D parameters. We evaluate both the validation loss and the mean friction R^2 score for the validation data which is shown in Fig. 7.6 together with the best epoch and the number of model parameters. Additionally, we evaluate the mean friction R^2 score for a selected set of configurations. This set consists of the top 10 configurations with respect to the max drop property for the Tetrahedron and Honeycomb patterns respectively. This is done as a way of evaluating the performance on the non-linear friction-strain curves which we find to be the more difficult trend to capture. The selected evaluation is shown in Fig. 7.7. Note that these configurations are already a part of the full dataset and thus the data points related to these configurations are most likely present in both the training and the validation data set. Hence, the performance must be considered in conjunction with the actual validation performance in Fig. 7.6.

From the validation scores in Fig. 7.6, looking at both the loss and the R^2 scores, we find that models $S(8-32)D(8-12)$ generally give the best performance. When considering at the best epoch, we find that

models of low depth result in a later best epoch in the range $\sim [800, 1000]$, in comparison to models of high depth yielding the best epoch in the range $\sim [300, 600]$. This indicates a transition from underfitting to overfitting of the model since the best validation scores are found earlier in the training process. However, since our training stores the best model during training, we do not have to worry too much about overfitting. Nonetheless, we can take this transition as a sign that our search is conducted in an appropriate complexity range. When consulting the evaluation on the selected sets in Fig. 7.7 we find significantly lower R^2 scores, especially for the Tetrahedron pattern. This observation indicates that the prediction of these configurations is more difficult, especially when considering that some of these data points are already included in the training data. While the peak R^2 value for the validation score in Fig. 7.6 was found for the model S16D10 model (98.23 %) the selected set test shows a slight preference for more complexity in the model. In the Tetrahedron selected set grid search, we find the best model to be S32D12, with an R^2 score of $\sim 87\%$. This model choice is more or less compatible with the overall performance since it is among the top candidates for the R^2 score and loss in Fig. 7.6 and the R^2 score for the selected Honeycomb set in Fig. 7.7 as well. Hence, we settle for this architecture.

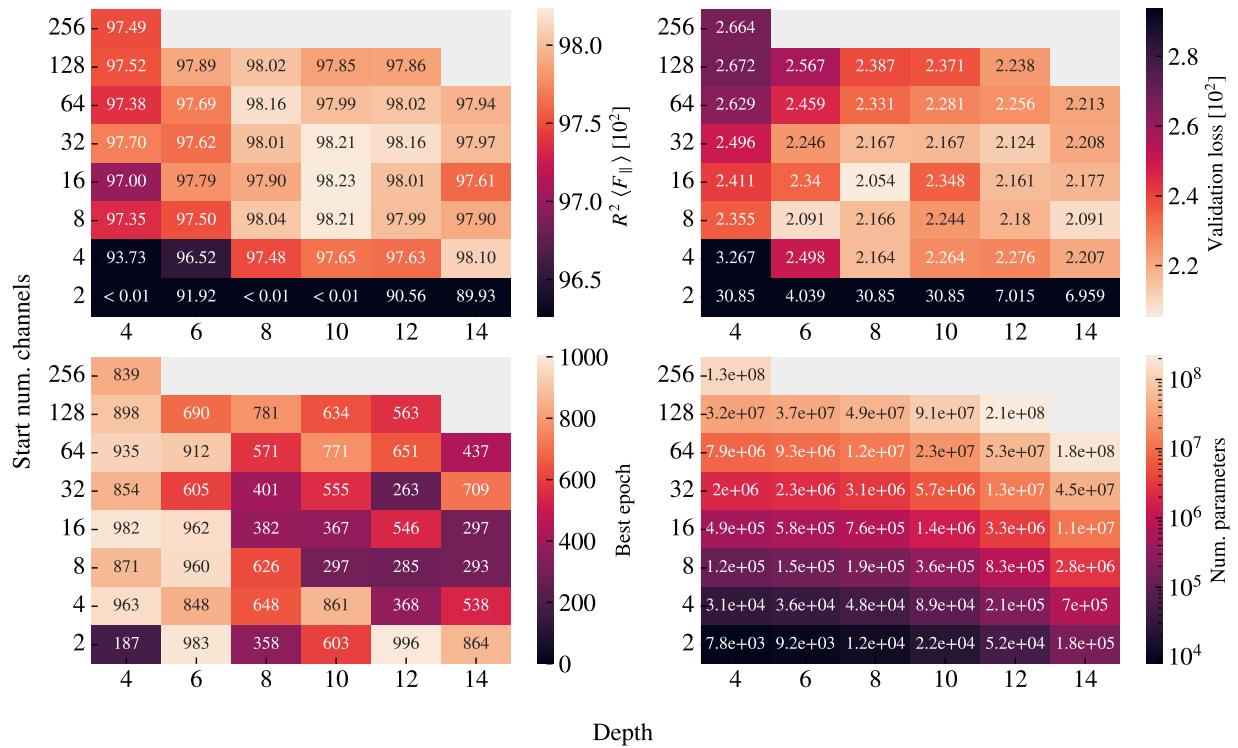


Figure 7.6: Architecture complexity grid search using a staircase-like VGGNet structure. The x-axis denotes the number of layers in the network (Depth) and the y-axis is the starting number of channels. See Eq. (7.2) for an example of the staircase architecture. For each architecture complexity, we evaluate the friction mean R^2 validation score (top left), the validation loss (top right), the best epoch stored based on validation scores (bottom left) and the number of model parameters (bottom right).

We note that the theoretical receptive field for the last convolutional layer (layer 6) is 13×13 according to Eq. (4.12), and thus each node in the first fully connected layer does not connect to the entirety of the 62×106 input image. Hence, some spatial dependence will have to be encoded in the fully connected part as well, and we note that an enlargement of the receptive field might serve as an interesting suggestion for an improvement of the model in further studies. One possible method is to increase the stride or use dilated convolutions.

Next, we consider momentum m and weight decay λ in the range $m \in [0.85, 0.99]$ and $\lambda \in [0, 10^{-2}]$. For each choice of momentum, we perform a learning rate range test. We propose two learning rate schemes: A constant learning rate scheme as used until this point and a one-cycle policy cyclic scheme. In the cyclic scheme, we set a maximum bound for which the learning rate starts from a factor 1/20 of the maximum bound, increases toward the maximum bound during the first 30% of training and decreases toward a factor 10^{-4} of the maximum bound for the remaining 70% of training. The increase and decrease are done by a cosine function. We let the momentum follow an inverse cycle with a minimum of $m = 0.80$ and a maximum corresponding to the momentum value being tested. The suggested learning rate for the constant learning



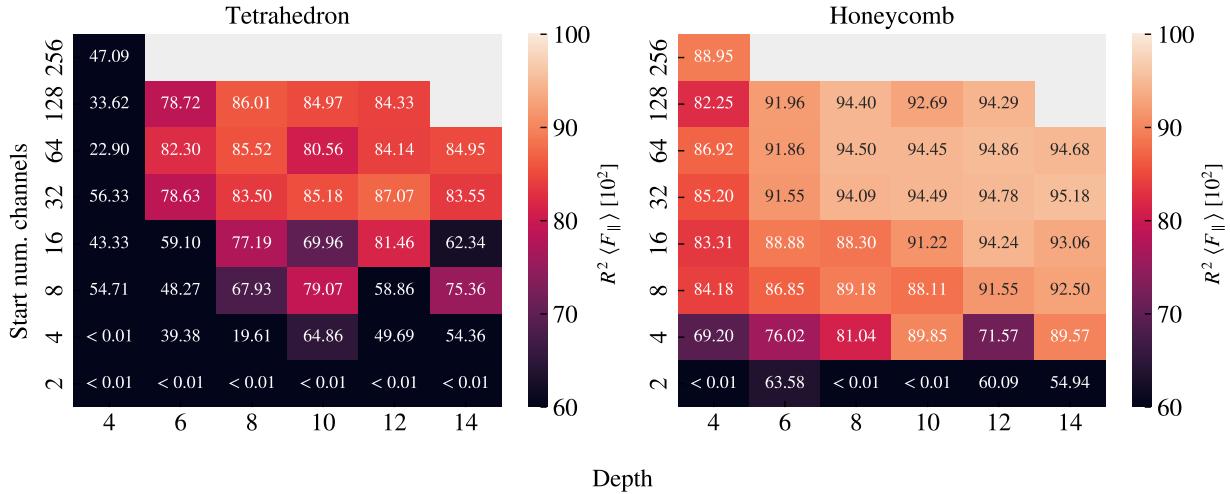


Figure 7.7: Architecture grid search similar to that of Fig. 7.6 on a selected set of configurations. The selected set consists of the top 10 candidates for the max drop property for the Tetrahedron and Honeycomb categories in the dataset respectively.

rate scheme is once again determined by the steepest slope on the learning range test loss curve while the maximum bound used for the cyclic scheme is determined as the point just before divergence. We find that the minimum point on the loss curve is a suitable choice that approaches the diverging point without getting too close and causing instabilities in the training. The learning rate range test for momentum is shown in Fig. 7.8. We observe generally that a higher momentum corresponds to a higher suggested learning rate for both schemes. Using these results we perform a grid search of momentum and weight decay. We examine again the validation loss and validation mean friction R^2 score in addition to the friction mean R^2 score for the selected set of Tetrahedron and Honeycomb patterns. This is shown for the constant learning rate scheme in Fig. 7.9 and for the cyclic scheme in Fig. 7.10.

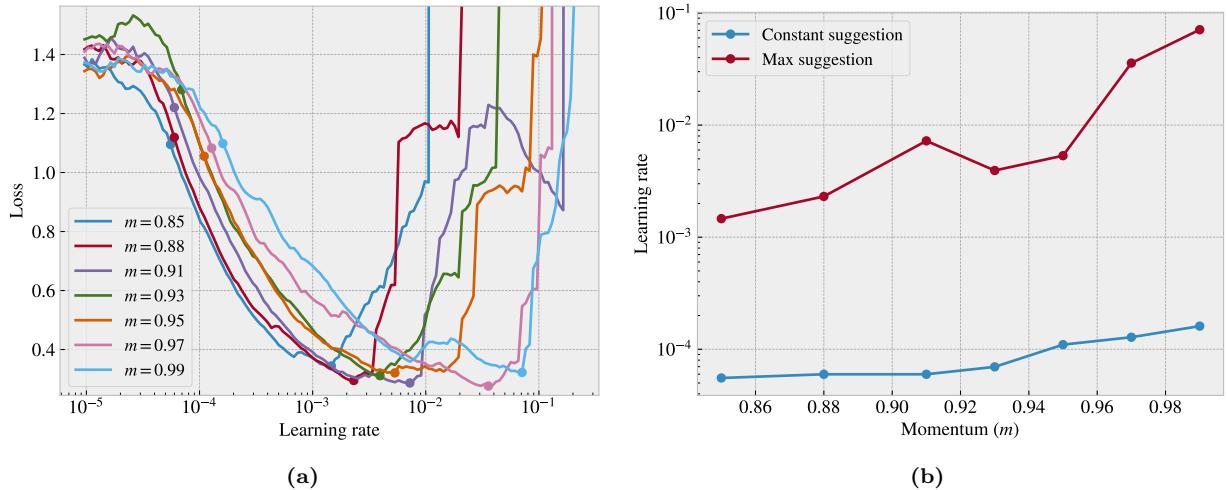


Figure 7.8: Learning rate range test for different momentum values m . (a) The validation loss for an exponentially increasing learning rate from 10^{-7} toward 10 with increments for each batch iteration during 1 epoch (yielding a total of 242 possible increments). As the curve diverges the test is halted. The dots on the validation loss vs. learning rate curve represent the steepest decline (at a lower learning rate) as an estimate for the constant learning scheme, and the minimum (at a higher learning rate) as an estimate for the maximum bound for the cyclic learning rate scheme. (b) The corresponding learning rate suggestion for the constant learning rate scheme and the maximum bound for the cyclic scheme respectively as a function of momentum choice.

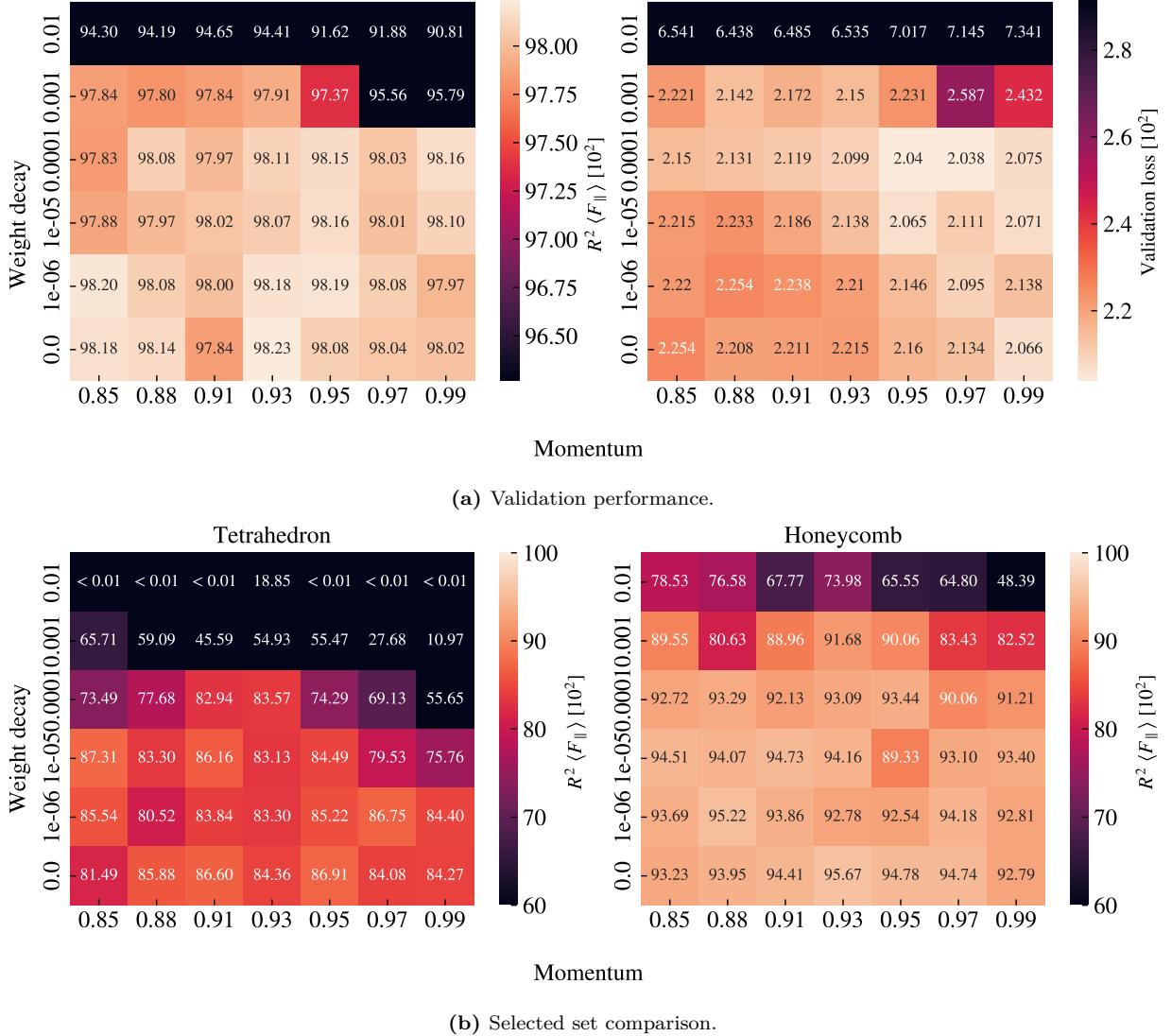
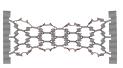


Figure 7.9: Momentum and weight decay grid search using a constant learning rate corresponding to the results from the learning rate range test in Fig. 7.8. (a) The friction mean R^2 validation score (left) and the validation loss (right). (b) The friction mean R^2 validation score for the selected set of Honeycomb (left) and Tetrahedron (right) patterns respectively, similar to that used in Fig. 7.7.



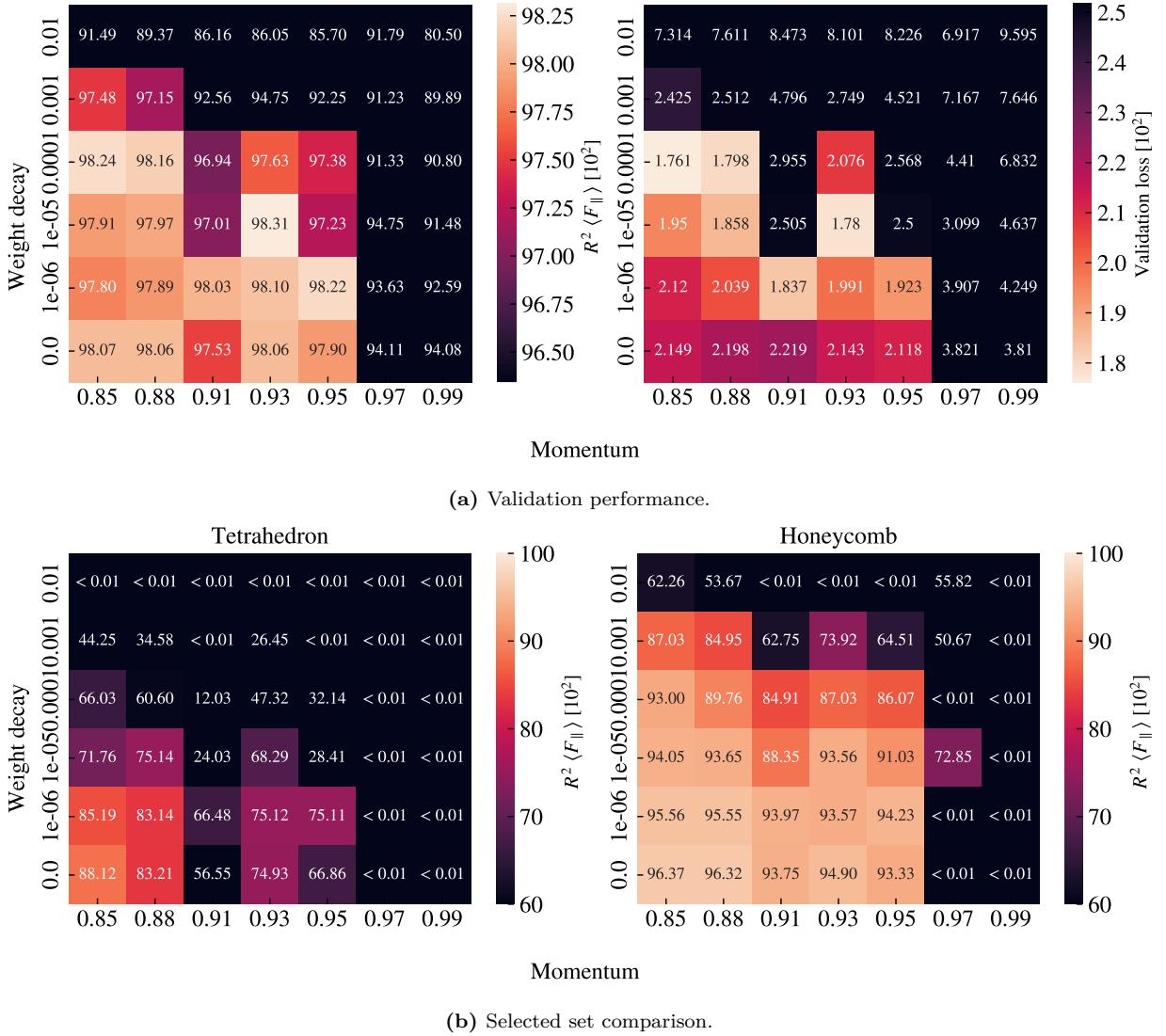


Figure 7.10: Momentum and weight decay grid search using a cyclic learning rate and cyclic momentum scheme. The learning rate maximum bound is chosen according to the learning rate range test in Fig. 7.8. The learning rate starts from a factor 1/20 of the maximum bound, increases toward the maximum bound during the first 30% of training and decreases toward a factor 10^{-4} of the maximum bound for the remaining 70% of training. This is done by following a cosine curve. The momentum performs an inverse cycling with the lowest momentum of 0.80 at the highest learning rate and a peak in momentum, corresponding to the values in the grid search, at the lowest learning rate. (a) The friction mean R^2 validation score (left) and the validation loss (right). (b) The friction mean R^2 validation score for the selected set of Honeycomb (left) and Tetrahedron (right) patterns respectively, similar to that used in Fig. 7.7.

The original validation scores, before varying momentum and weight decay, were a validation loss of 0.02124 and a mean friction R^2 score of 0.9816. By varying momentum and weight decay, we find that these scores can be improved slightly for the constant learning rate scheme (loss: 0.02038, R^2 : 0.9823) and even more for the cyclic scheme (loss: 0.0176, R^2 : 0.9831). Note that the loss and R^2 scores here do not correspond to the same hyperparameter choices. The comparison among best scores is summarized in Table 7.3. In general, the constant scheme shows rather stable results for all momentum settings $m \in [0.85, 0.99]$ in combination with a low weight decay $\lambda \leq 10^{-4}$. For the cyclic scheme the performance peaks toward a low momentum $m \leq 0.93$ and a low weight decay $\lambda \leq 10^{-4}$. Looking at the summary in Table 7.3, we see that the cyclic scheme can produce a high score among all four performance metrics, but since these scores do not share common hyperparameters we need to choose which of them to prioritize. Due to our interest in capturing the non-linear trends, we prioritize the score from the selected set of Tetrahedron patterns as this has proven to be the greatest challenge for our model to capture. We recognize that this choice introduces a greater risk of overfitting since the data points within this evaluation set are partly included in the training

set as well. This is especially alarming since the absence of weight decay allows for more overfitting in general. However, for the purpose of performing an accelerated search, we find it more important to increase the likelihood of discovering novel designs than to reduce the risk of getting false positive results. Since we retain the option to verify the properties of a given design through MD simulations afterward, we do not have to rely on the machine learning prediction indefinitely. Thus we choose the cyclic trained model with low maximum momentum $m = 0.85$ and zero weight decay as our final model. On a final note, we also point out that our choice of hyperparameters corresponded to the edge of our grid search. Thus, it would have been natural to perform an extended search in that range, but due to time constraints and the belief that the potential gain from doing so was not significant, we decided to omit it.

Table 7.3: The best validation loss and R^2 scores from the momentum and weight grid search using the S32D12 model. We compare the scores from the constant and cycling learning rate and momentum scheme against the original scores from the S32D12 model in the architecture complexity grid search. The scores correspond to different hyperparameter choices that maximize each score respectively.

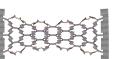
| | | Score [10 ²] | Momentum | Weight decay |
|-------------------|----------|--------------------------|----------|------------------|
| Validation loss | Original | 2.124 | 0.9 | 0 |
| | Constant | 2.038 | 0.97 | 10 ⁻⁴ |
| | Cyclic | 1.761 | 0.85 | 10 ⁻⁴ |
| Validation R^2 | Original | 98.16 | 0.9 | 0 |
| | Constant | 98.23 | 0.93 | 0 |
| | Cyclic | 98.31 | 0.93 | 10 ⁻⁵ |
| Tetrahedron R^2 | Original | 87.07 | 0.9 | 0 |
| | Constant | 87.31 | 0.85 | 10 ⁻⁵ |
| | Cyclic | 88.12 | 0.85 | 0 |
| Honeycomb R^2 | Original | 94.78 | 0.9 | 0 |
| | Constant | 95.67 | 0.93 | 0 |
| | Cyclic | 96.37 | 0.85 | 0 |

7.4.5 Model performance

From the hypertuning process, we settled on the S32D12 model trained by a cyclic scheme with a maximum momentum of 0.85, a maximum learning rate bound of 0.015 chosen accordingly to the learning range test in Fig. 7.8 and zero weight decay. The model contains 1.3×10^7 model parameters. The main performance metrics are shown in Table 7.4 where ‘‘Tetrahedron’’ and ‘‘Honeycomb’’ refer to the selected set scores. Although we have mainly considered the mean friction R^2 score during the hypertuning we find that the performance on the remaining parameters is reasonable as well. The validation set reveals a final R^2 score for the mean friction of $\sim 98\%$ and a rupture accuracy of $\sim 96\%$. Since the data only contains roughly 12% ruptures this should be compared to a score of 88% corresponding to simply predicting no ruptures at all. We find a considerably large relative error for the rupture strain of $\sim 13\%$. However, this error is lower for the Tetrahedron (5.9%) and Honeycomb (1.5%) sets. Hence, it is possible to infer that the high relative error in the validation set is due to some instances of very low rupture strains that might shift up the average value for the relative error. Fig. 7.11 shows the mean friction, max friction and relative contact predictions for the Tetrahedron (7, 5, 1) and Honeycomb (2, 2, 1, 5) patterns used in the pilot study. We note that these configurations are also partly contained in the training data, but this serves as a way of comparing the prediction quality estimated for the R^2 scores with a visual evaluation. Later on, we will evaluate a true test set based on the proposals from the accelerated search.

Table 7.4: Evaluation of the final model performance considering all the model outputs. Mean values are taken over the scores for each configuration within the categories: The validation set, the selected Tetrahedron set and the selected Honeycomb set.

| | Loss [10 ²] | R^2 [10 ²] | | | Abs. [10 ²] | Rel. [10 ²] | Acc. [10 ²] |
|-------------|-------------------------|--------------------------|-----------|---------|-------------------------|-------------------------|-------------------------|
| | Total | Mean F_f | Max F_f | Contact | Porosity | Rup. Strain | Rupture |
| Validation | 2.1488 | 98.067 | 93.558 | 94.598 | 2.325 | 12.958 | 96.102 |
| Tetrahedron | 4.0328 | 88.662 | 85.836 | 64.683 | 1.207 | 5.880 | 99.762 |
| Honeycomb | 8.6867 | 96.627 | 89.696 | 97.171 | 1.040 | 1.483 | 99.111 |



With the use of our final model, we evaluate the performance for the task of ranking the configurations by the properties of interest. That is, we go through all the configurations in the dataset, for the Tetrahedron, Honeycomb and Random walk respectively, calculate the properties of interest and sort the configurations accordingly. This is shown in Table 7.5 in comparison to the actual ranking in the dataset. Generally, we find that the ML model performs rather well in the ranking of the maximum properties getting the right configurations into the top 3 three, while it is performing a lot worse for the minimum friction property. This latter observation can be attributed to the fact that the precision needed for an accurate ranking among the minimum friction cases is a lot higher than for the remaining properties. This lack of precision is especially highlighted by the fact that the model predicts the 12th Random walk pattern (see Table 7.5c) to have negative mean friction, which is clearly outside the bound of reasonable values. For the maximum categories, we find that the model gives a slightly better ranking for the Tetrahedron and Honeycomb in comparison to the Random walk patterns. When considering the actual predicted property scores for the maximum properties we find that the model predictions are generally within a ~ 0.2 nN deviation in the top 5. This supports that our model can be used to perform an accelerated search of new configurations yielding a meaningful ranking of property scores.

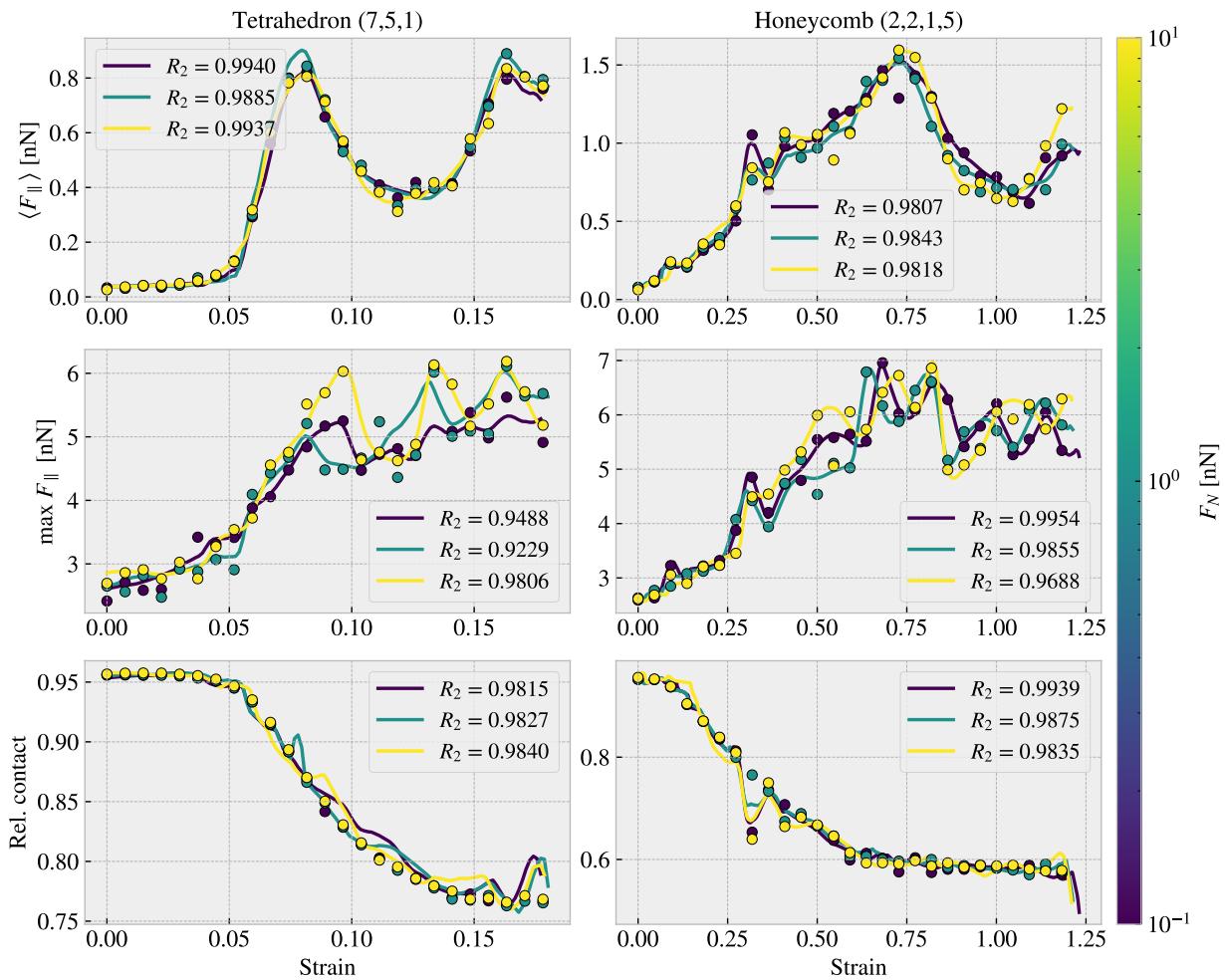


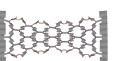
Figure 7.11: Visual evaluation of the final model predictions on the Tetrahedron (7, 5, 1) and Honeycomb (2, 2, 1, 5) used in the pilot study, for the mean friction $\langle F_{\parallel} \rangle$, maximum friction $\text{max } F_{\parallel}$ and the relative contact as a function of strain. The model predictions (solid line) are based on 10^3 data points in the strain range $[0, 1.5]$, with the curve being cut off after the first rupture prediction. This is compared to the data points (dots). The color denotes the corresponding normal loads. The R^2 scores are shown for each prediction fit for each load value.

| ML Rank | Data | | ML | | Data Rank |
|-------------------------------|-----------|------------|-----------|------------|--------------|
| | Config | Value [nN] | Config | Value [nN] | |
| $\min F_{\text{fric}}$ | | | | | |
| 20 | (3, 9, 4) | 0.0067 | (3, 1, 2) | 0.0041 | 5 |
| 5 | (3, 1, 3) | 0.0075 | (1, 3, 4) | 0.0049 | 11 |
| 6 | (5, 3, 4) | 0.0084 | (1, 3, 3) | 0.0066 | 6 |
| 21 | (1, 7, 3) | 0.0084 | (3, 1, 4) | 0.0066 | 8 |
| 1 | (3, 1, 2) | 0.0097 | (3, 1, 3) | 0.0078 | 2 |
| $\max F_{\text{fric}}$ | | | | | |
| 1 | (5, 3, 1) | 1.5875 | (5, 3, 1) | 1.5920 | 1 |
| 2 | (1, 3, 1) | 1.4310 | (1, 3, 1) | 1.2739 | 2 |
| 4 | (3, 1, 2) | 1.0988 | (9, 3, 1) | 1.1162 | 4 |
| 3 | (9, 3, 1) | 1.0936 | (3, 1, 2) | 0.7819 | 3 |
| 5 | (7, 5, 1) | 0.7916 | (7, 5, 1) | 0.7740 | 5 |
| $\max \Delta F_{\text{fric}}$ | | | | | |
| 1 | (5, 3, 1) | 1.5529 | (5, 3, 1) | 1.5578 | 1 |
| 2 | (1, 3, 1) | 1.3916 | (1, 3, 1) | 1.2331 | 2 |
| 4 | (3, 1, 2) | 1.0891 | (9, 3, 1) | 1.0807 | 4 |
| 3 | (9, 3, 1) | 1.0606 | (3, 1, 2) | 0.7778 | 3 |
| 5 | (7, 5, 1) | 0.7536 | (7, 5, 1) | 0.7399 | 5 |
| $\max \text{drop}$ | | | | | |
| 1 | (5, 3, 1) | 0.8841 | (5, 3, 1) | 0.8603 | 1 |
| 2 | (3, 5, 1) | 0.4091 | (3, 5, 1) | 0.3722 | 2 |
| 4 | (7, 5, 1) | 0.3775 | (1, 1, 1) | 0.2879 | 5 |
| 5 | (9, 7, 1) | 0.2238 | (7, 5, 1) | 0.2478 | 3 |
| 3 | (1, 1, 1) | 0.1347 | (9, 7, 1) | 0.1302 | 4 |

(a) Tetrahedron ranking.

| ML Rank | Data | | ML | | Data Rank |
|-------------------------------|--------------|------------|--------------|------------|--------------|
| | Config | Value [nN] | Config | Value [nN] | |
| $\min F_{\text{fric}}$ | | | | | |
| 1 | (2, 5, 1, 1) | 0.0177 | (2, 5, 1, 1) | 0.0113 | 1 |
| 9 | (2, 4, 5, 1) | 0.0187 | (2, 5, 5, 3) | 0.0149 | 7 |
| 7 | (2, 4, 1, 1) | 0.0212 | (2, 5, 5, 1) | 0.0182 | 4 |
| 3 | (2, 5, 5, 1) | 0.0212 | (2, 5, 3, 1) | 0.0186 | 5 |
| 4 | (2, 5, 3, 1) | 0.0226 | (2, 4, 1, 3) | 0.0198 | 15 |
| $\max F_{\text{fric}}$ | | | | | |
| 1 | (2, 1, 1, 1) | 2.8903 | (2, 1, 1, 1) | 2.9171 | 1 |
| 2 | (2, 1, 5, 3) | 2.2824 | (2, 1, 5, 3) | 2.4004 | 2 |
| 6 | (2, 1, 3, 1) | 2.0818 | (2, 1, 5, 1) | 2.1060 | 5 |
| 4 | (2, 1, 3, 3) | 2.0313 | (2, 1, 3, 3) | 1.9458 | 4 |
| 3 | (2, 1, 5, 1) | 2.0164 | (2, 4, 1, 1) | 1.9381 | 6 |
| $\max \Delta F_{\text{fric}}$ | | | | | |
| 1 | (2, 1, 5, 3) | 2.0234 | (2, 1, 5, 3) | 2.1675 | 1 |
| 2 | (2, 1, 1, 1) | 1.9528 | (2, 1, 1, 1) | 2.0809 | 2 |
| 3 | (2, 4, 1, 1) | 1.8184 | (2, 4, 1, 1) | 1.9157 | 3 |
| 4 | (2, 1, 3, 3) | 1.7645 | (2, 1, 3, 3) | 1.6968 | 4 |
| 5 | (2, 4, 1, 3) | 1.4614 | (2, 4, 1, 3) | 1.5612 | 5 |
| $\max \text{drop}$ | | | | | |
| 1 | (2, 3, 3, 3) | 1.2785 | (2, 3, 3, 3) | 1.3642 | 1 |
| 2 | (2, 1, 3, 1) | 1.1046 | (2, 1, 3, 1) | 0.9837 | 2 |
| 3 | (2, 3, 3, 5) | 0.8947 | (2, 3, 3, 5) | 0.9803 | 3 |
| 4 | (2, 1, 5, 3) | 0.8638 | (2, 1, 5, 3) | 0.9556 | 4 |
| 13 | (2, 5, 1, 1) | 0.8468 | (2, 4, 5, 3) | 0.8999 | 8 |

(b) Honeycomb ranking.

Table 7.5: (Table continues on the next page)

| ML Rank | Data | | ML | | Data Rank |
|-------------------------------|--------|------------|--------|------------|--------------|
| | Config | Value [nN] | Config | Value [nN] | |
| $\min F_{\text{fric}}$ | | | | | |
| 1 | 12 | 0.0024 | 12 | -0.0011 | 1 |
| 24 | 76 | 0.0040 | 06 | 0.0036 | 27 |
| 6 | 13 | 0.0055 | 14 | 0.0074 | 23 |
| 31 | 08 | 0.0065 | 05 | 0.0082 | 19 |
| 26 | 07 | 0.0069 | 63 | 0.0085 | 57 |
| $\max F_{\text{fric}}$ | | | | | |
| 3 | 96 | 0.5758 | 99 | 0.5155 | 2 |
| 1 | 99 | 0.5316 | 98 | 0.4708 | 3 |
| 2 | 98 | 0.4478 | 96 | 0.4356 | 1 |
| 4 | 97 | 0.3624 | 97 | 0.3503 | 4 |
| 11 | 58 | 0.3410 | 55 | 0.2817 | 7 |
| $\max \Delta F_{\text{fric}}$ | | | | | |
| 3 | 96 | 0.5448 | 99 | 0.4669 | 2 |
| 1 | 99 | 0.4769 | 98 | 0.4314 | 3 |
| 2 | 98 | 0.4085 | 96 | 0.4128 | 1 |
| 4 | 97 | 0.3268 | 97 | 0.3080 | 4 |
| 78 | 57 | 0.2978 | 55 | 0.2542 | 7 |
| $\max \text{drop}$ | | | | | |
| 3 | 01 | 0.1818 | 00 | 0.1883 | 3 |
| 2 | 96 | 0.1733 | 96 | 0.1654 | 2 |
| 1 | 00 | 0.1590 | 01 | 0.1532 | 1 |
| 11 | 37 | 0.1022 | 04 | 0.0591 | 8 |
| 28 | 34 | 0.0879 | 56 | 0.0552 | 20 |

(c) Random walk ranking.

Table 7.5: Ranking of the dataset according to the four properties of interest using the final machine learning (ML) model for the Tetrahedron (a), Honeycomb (b) and Random walk (c) patterns in the dataset respectively. The ranking is shown in descending order for each section of rows corresponding to the four properties of interest. The left side of the vertical center line denotes the true data ranking showing the top 5 scores in descending order (the top row shows rank 1 and the bottom row shows rank 5). The outermost left column (ML rank) then denotes the corresponding ranks given by the ML model. The right side of the vertical center line shows the top 5 ranking given by the ML model for which the outermost right column shows the corresponding true data ranks. If the model gets the top 5 ranking right both the outermost left and right columns show 1, 2, 3, 4, 5 in descending order.

7.5 Accelerated Search

From Sec. 7.4 we have found promising results that we can use our machine learning model to predict the frictional behavior of a Kirigami sheet. This enables us to omit the MD simulations in the evaluation process and perform an accelerated search through new configurations. We will use the friction properties of interest as our main metrics for optimization. We approach the accelerated search by two different methods:

1. Using the generative algorithms developed for the creation of the Tetrahedron, Honeycomb and Random walk patterns, we create an extended dataset and evaluate the performance using the ML model.
2. Using a genetic algorithm method we perturb (mutate) the configurations and optimize for the max drop property using the ML model to evaluate the fitness function.

7.5.1 Patteren generation search

We utilize the pattern generators developed in Chapter 5 to create an extended dataset for our search. For the Tetrahedron and Honeycomb patterns, the increments of the parameters will eventually lead to the main pattern structures exceeding the size of the sheet. Thus, we can essentially perform a full search “maxing out” the parameters of these patterns. We estimate that this is done with the maximum parameters, (60, 60, 30) for the Tetrahedron, and (30, 30, 30, 60) for the Honeycomb pattern. We use a random reference position and regenerate each unique parameter set 10 times to explore translational effects. This gives in total 1.35×10^5

configurations for the Tetrahedron pattern and 2.025×10^6 for the Honeycomb pattern. For the Random walk generator, we perform a Monte Carlo sampling. That is, in each sample we draw the scalar values, either from a uniform (U) or logarithmic uniform (LU) distribution as follows.

$$\begin{array}{lll} \text{Num. walks, } \sim U[1, 30], & \text{Max. steps, } \sim U[1, 30], & \text{Min. dis., } \sim U[0, 4], \\ \text{Bias direction, } \sim U[0, 2\pi], & \text{Bias. strength, } \sim LU[0, 10], & p_{\text{stay}}, \sim U[0, 1]. \end{array}$$

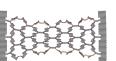
Notice that we use a discrete distribution for the parameters requiring integers. For the binary parameters *connection*, *avoid invalid*, *RN6* and *grid start* we simply set the values by a 50–50 chance. The remaining parameters are kept constant at *periodic* set to true and *centering* set to false throughout the search. For the handling of clustering, we implement the extended repair algorithm such that the sheet is repaired by the least modifications approach rather than retrying the generation several times. Due to the extra computation time associated with the random walk and the repair algorithm, we only generate 10^4 configurations within this class. For the ML evaluation of the generated configurations we use a normal load of 5 nN and generate a friction-strain curve in the strain domain [0, 2] using 100 uniformly spaced points. We compute the properties of interest and rank the configurations accordingly. The top candidate scores for each property are shown in Table 7.6 including a comparison to the original dataset top candidates (from Table 7.2). The random walk top five candidates for each property respectively are visualized in Fig. 7.12.

Table 7.6: Results for the accelerated search using the pattern generators. The top search candidates for each of the four properties of interest are shown in the left section (Search) regarding the Tetrahedron, Honeycomb and Random walk patterns respectively. The upper rows show the scores and the lower rows the associated names (parameters). The right section (Data) shows the corresponding scores from the best candidates within the dataset (from Table 7.2). All scores are given in units nN.

| Scores | Search | | | Data | | |
|------------------------------|--------------|-----------------|----------------|-------------|--------------|-------------|
| | Tetrahedron | Honeycomb | Random walk | Tetrahedron | Honeycomb | Random walk |
| min F_{fric} | -0.062 | -0.109 | -0.061 | 0.0067 | 0.0177 | 0.0024 |
| max F_{fric} | 1.089 | 2.917 | 0.660 | 1.5875 | 2.8903 | 0.5758 |
| max ΔF_{fric} | 1.062 | 2.081 | 0.629 | 1.5529 | 2.0234 | 0.5448 |
| max drop | 0.277 | 1.250 | 0.269 | 0.8841 | 1.2785 | 0.1818 |
| Configs. | | | | | | |
| Configs. | Tetrahedron | Honeycomb | Random walk | Tetrahedron | Honeycomb | Random walk |
| min F_{fric} | (13, 11, 14) | (14, 25, 7, 19) | <i>no name</i> | (3, 9, 4) | (2, 5, 1, 1) | 12 |
| max F_{fric} | (1, 3, 1) | (2, 1, 1, 1) | <i>no name</i> | (5, 3, 1) | (2, 1, 1, 1) | 96 |
| max ΔF_{fric} | (1, 3, 1) | (2, 1, 1, 1) | <i>no name</i> | (5, 3, 1) | (2, 1, 5, 3) | 96 |
| max drop | (1, 7, 1) | (3, 3, 5, 3) | <i>no name</i> | (5, 3, 1) | (2, 3, 3, 3) | 01 |

First of all, we notice that the top candidates for the minimum friction are all predicted to have a negative friction value. This unphysical prediction aligns with the previous observations that our model does not have the required precision to yield accurate predictions for this property. Moreover, we can argue that pursuing the optimization for a low friction value will eventually exploit the weaknesses of the model as we reward an unphysical negative value. In order to resolve this problem it may be necessary to extend the training dataset and possibly include a physical constraint for positive friction values. However, by consulting the proposed minimum candidates we find that they all share the same feature of being sparsely cut. For the Random walk, we see this visually in Fig. 7.12, while for the Tetrahedron and Honeycomb patterns, this is evident from the configuration parameters shown in Table 7.6 where the parameters reveal a high spacing between the cuts. The porosity of the minimum friction top candidates are all rather low being 1.5%, 5.6%, and 1.6% for the Tetrahedron, Honeycomb and Random walk respectively. This further supports the idea that the Kirigami sheet can not readily be used to reduce friction (within our system domain) since the results point toward a non-cut sheet as the best minimum friction candidate.

Among the remaining maximum properties, we find competing scores with the Honeycomb and Random walk classes. However, the top-scoring values for the Honeycomb candidates correspond to configurations already within the original dataset, which is also the case for the Tetrahedron top candidates. The only difference is the randomized reference position making for a translated version of the pattern. When taking a closer look at the full ranking for each property it becomes apparent that the predictions are highly sensitive to the reference position parameter used for the Tetrahedron and Honeycomb pattern. Since we repeated the pattern generation 10 times for each parameter set with a random reference position, we initially expected to get a ranking in sets of 10. However, the ranking only shows contiguous appearing sets in the range of 1–5 which points toward a dependency on pattern translation. Hence we investigate this further by evaluating the



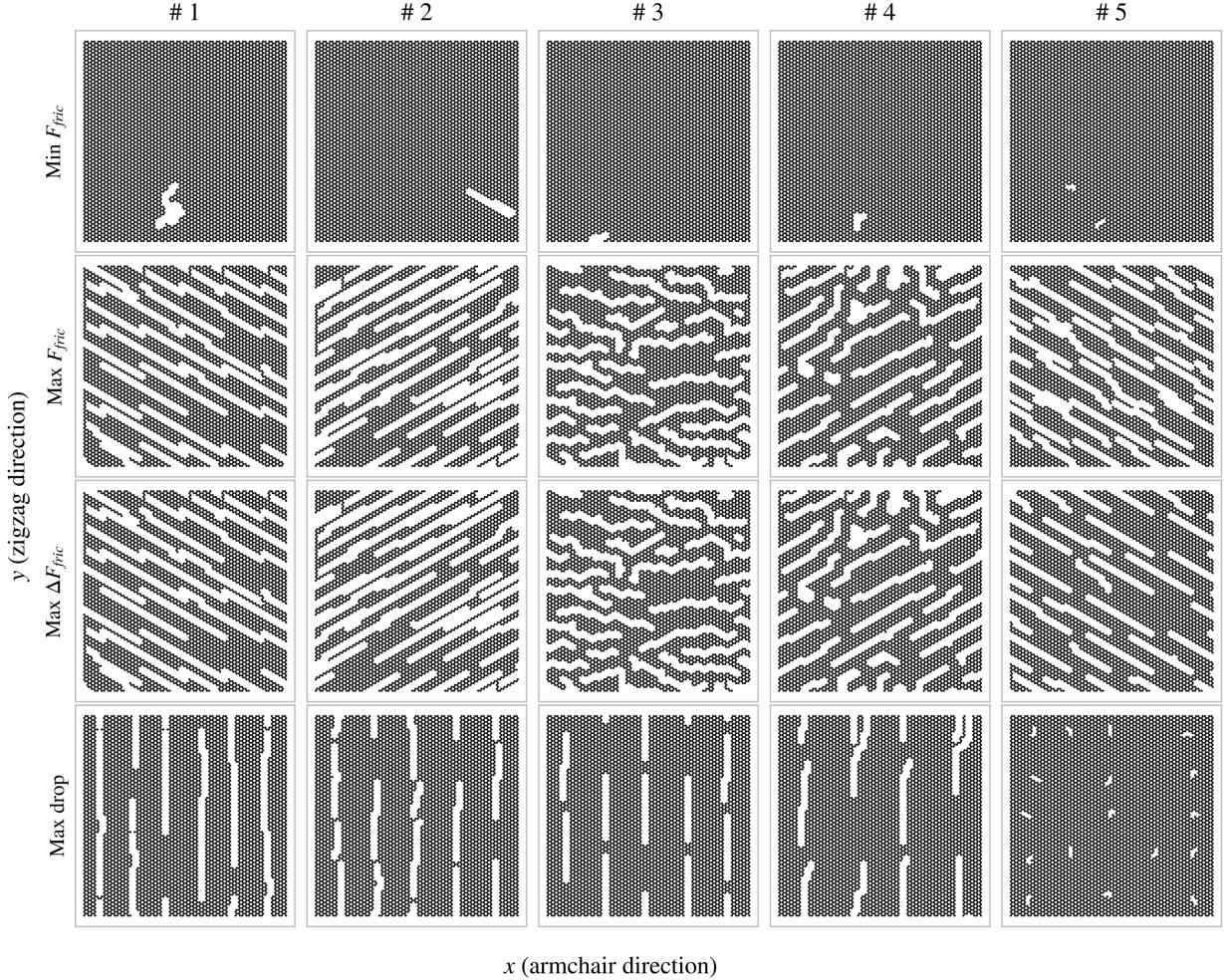


Figure 7.12: Top 5 candidates for the accelerated search using the Random walk generator. The rows denote the four properties of interest and the columns the top 5 candidates found in the search corresponding to descending scores from left to right.

scores for a systematic change of the reference position. We generally find the max drop parameter to give the highest variation and thus we show the max drop scores for the max drop top candidates: Tetrahedron (1, 7, 1), (5, 3, 1) and Honeycomb (3, 3, 5, 3), (2, 3, 3, 3) in Fig. 7.13. The results show that the max drop property prediction varies drastically with the translation of these patterns. The emerging question is then whether this is grounded in a physical phenomenon or simply a deficiency in the model. Even though the patterns are periodic in the x-y-plane, with a period according to the unique number of translations shown in Fig. 7.13, the translation will determine the specific configuration of the edge. Previous studies of static friction and stick-slip behavior point to the importance of edge effects [90], and thus for a sheet where the atoms on the free edge (in the $\pm x$ direction) constitute about 2.5% of the inner sheet atom count, it is not unreasonable that the translation might result in a significantly different outcome. In that case, the search through reference positions highlights that the translation can be key to optimizing for certain properties. However, the results might also indicate that the model is either overfitted or that we simply did not provide enough data to reach a generalization of the complex physical behavior of the system. The sensible way forward to unravel this would be to perform additional MD simulations for translational variants of the same configurations to investigate for any physical edge dependencies or otherwise strengthen the model by this data. We leave this suggestion for another study. When considering some of the friction-strain curves corresponding to the result in Fig. 7.13 we also find that the prediction of the rupture point plays an important role in the max drop property score. Since the rupture is often predicted on a descending part of the curve any variation to the rupture strain will affect the max drop property quite significantly.

In order to get more insight into the generalization of the model, we evaluate the performance on a true test set. We use the 20 configurations given by the top 5 Random walk candidates for each of the properties of interest shown in Fig. 7.12. We calculate the ground truth data using MD simulations with 30 strain values uniformly spaced within the rupture strain and a normal load of 5 nN. Unfortunately, the test set reveals a

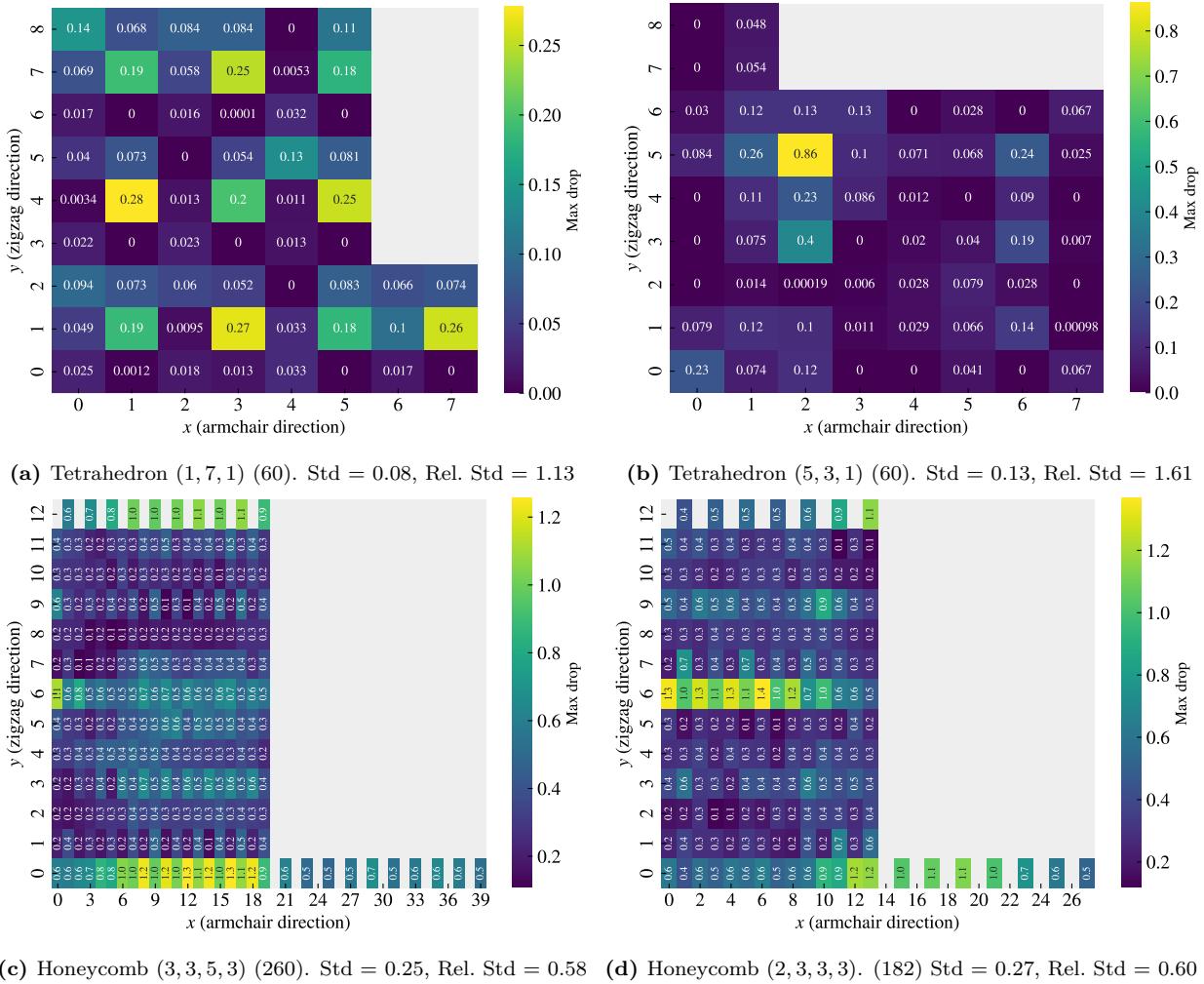
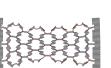


Figure 7.13: Prediction of the max drop property for selected patterns using the machine learning model for all unique reference positions. The heatmap and the annotated values denote the property score with respect to the reference position x and y coordinate. Panels (a) to (d) show the results from various patterns which correspond to the candidates for the max drop property in the accelerated pattern search and the dataset respectively. The sub-caption states the pattern parameters, the number of total unique reference positions in parenthesis, the standard deviation and the standard deviation relative to the mean score.

substantially worse performance than the validation scores reported so far. It shows a loss of 2.13, which is two orders of magnitude higher than the validation loss, an average absolute error for the mean friction of 0.14 nN and a rupture accuracy of 70 %. The corresponding mean friction average R^2 score is negative which indicates that our model performs worse than simply guessing on the true data mean. This reveals that our model is not generalized enough to provide accurate predictions on the newly generated Random walk configurations. This can mainly be attributed to two reasons: 1) The test set data distribution is not similar to that of the training and validation data drawn from the original data set. 2) The considerations of the selected Tetrahedron and Honeycomb dataset, which overlapped with the training data, has led to an overfitting of the model in the hypertuning. In order to test the last hypothesis we went back to the beginning of the hypertuning process and chose the best model (C16D8) based purely on validation loss in the architecture complexity grid test in Fig. 7.6. By using this model on the test set we find similar poor results which suggests that the test performance issue is not caused by the hypertuning process. Instead, it points to the fact that our original training data does not contain a generalized enough configuration distribution to accurately capture the full complexity of our system. This aligns with the high fluctuation in prediction value when translating the patterns. Thus we conclude that a machine learning approach might be feasible, given the promising validation scores, but that we need a bigger and more generalized dataset for a reliable prediction of new configurations.



7.5.2 Genetic algorithm search

Although our machine learning analysis indicates that the model is not generalized enough for an accurate prediction on new configurations, we carry out a short investigation on the use of a genetic-algorithm-based accelerated search. So far we have concluded that a minimization of the friction is not promising, and hence we discard this property for further study. We have also seen that the maximum style properties often share similar top candidates, and thus we choose to only investigate the max drop property associated with the aim of achieving a negative friction coefficient in the coupled system.

In order to verify our implementation of the genetic algorithm described in Sec. 4.6, we consider a smaller 10×10 square lattice initially. We choose to evaluate the energy of a zero-temperature Ising model without any external field. That is, we consider the system described by the Hamiltonian

$$H = -J \sum_{\langle kl \rangle} s_k s_l, \quad (7.3)$$

where s denotes the value of each lattice site and the sum is running over all nearest neighbor pairs in the lattice. The sites take a binary value, either -1 or $+1$, meaning that the lowest energy (-200) is reached when all site values have an equal sign, all -1 or all $+1$. The highest energy (200) corresponds to a checkerboard pattern where each site has nearest neighbors with an opposite sign to its own. We consider periodic boundary conditions, meaning that the sites on the edge will be connected to the sites on the opposite edge. We create a small population of 10 individuals from a random noise initialization where each site has a probability of p being -1 and a probability $p - 1$ being $+1$. Hence, the probability represents the average porosity of the configurations and we choose $p = 0.5$ as an unbiased choice. When considering the minimization of energy, the algorithm converges relatively fast (within tens of generations) to the best possible score. Thus, we consider the more challenging problem of maximizing energy which requires the configuration to reach the checkerboard pattern. Fig. 7.14 shows the score for a maximization of the energy using the genetic algorithm. We observe that the score converges faster initially, but that it eventually reaches the best score at generation 257. This indicates that the algorithm can handle an optimization problem with some level of spatial dependency. However, we note that some initializations were resulting in a saturating convergence, where two unsynchronized checkerboard patterns formed in distinct regions of the lattice. This leads to a local maximum because transitioning from two unsynchronized checkerboard patterns to a single synchronized pattern would necessitate a slight temporary decrease in the score.

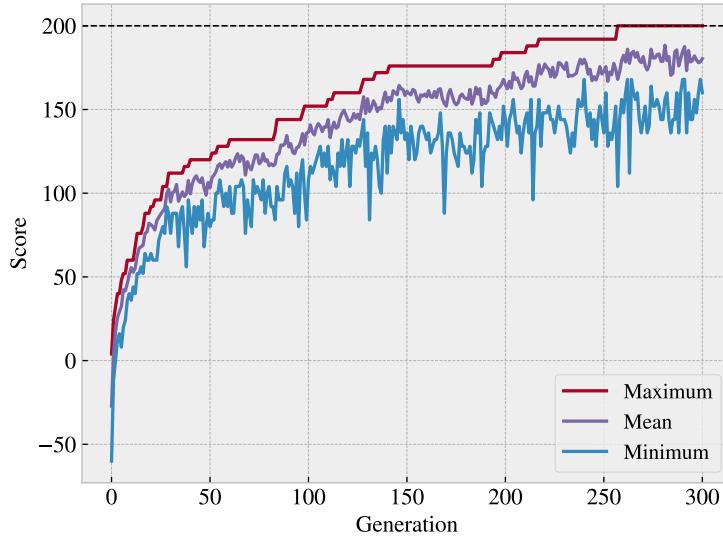


Figure 7.14: Optimization for the maximum Ising energy given by Eq. (7.3) for a 10×10 lattice using a population size of 10. The population was initialized with random noise having a probability of 0.5 being -1 and a probability of 0.5 being $+1$. The three curves indicate the minimum, mean, and maximum scores in the population. The best score indicated by the dotted line is reached at generation 257.

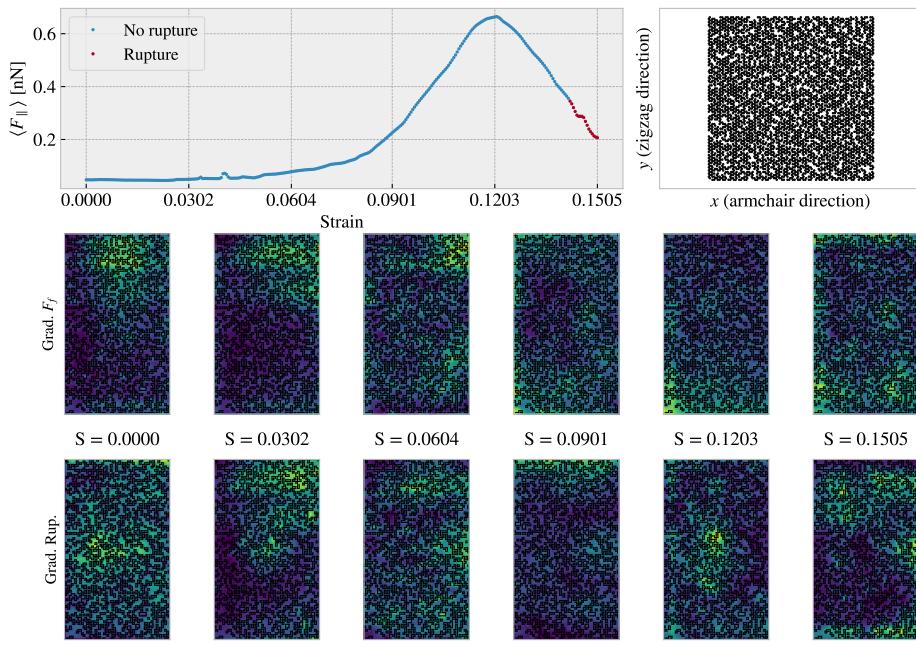
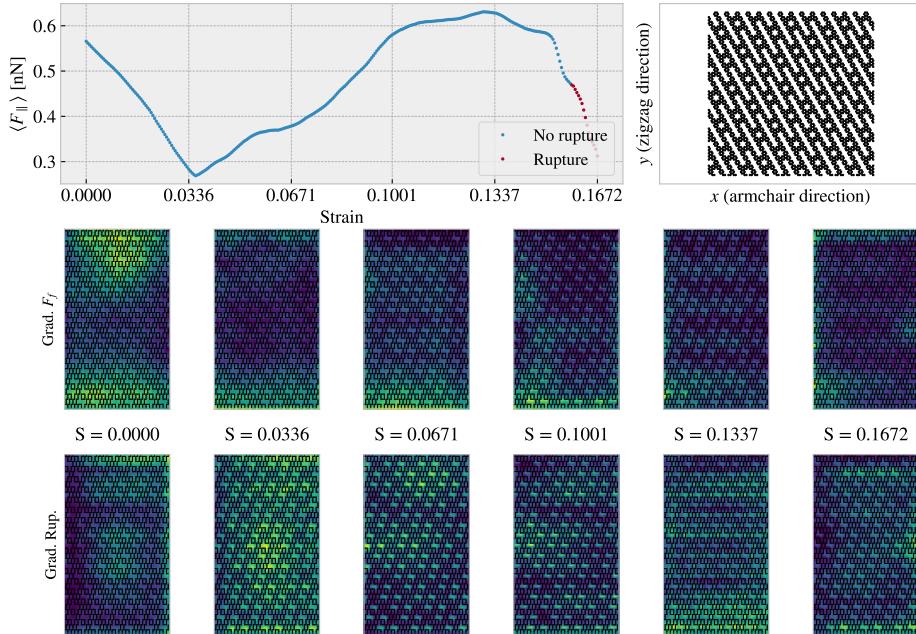
Moving on from the verification of the algorithm, we consider the optimization for the max drop property for the Kirigami patterns. We utilize the machine learning model to evaluate the friction-strain curve for each pattern and compute the corresponding max drop property value, similar to what we have done in Sec. 7.5.1. For the initialization of the population, we take a basis in the top candidates for the pattern generation search from Sec. 7.5.1. That is, we generate new configurations using the parameters that led to the highest

score for the max drop property. We do this for the top candidates within the Tetrahedron, Honeycomb and Random walk categories respectively. We generate a population of 100 configurations and run the search for 100 generations since we did not see much improvement for longer runs.

We find immediately that the Tetrahedron and Honeycomb search did not give any useful results. In both cases, the highest-scoring individual in the population at generation 0 was still the best candidate at the end of the search, even though the average score was rising initially. For the Random walk, we perform 5 runs for different population initializations. We select the top 5 max drop candidates from the Random walk pattern search (seen in the bottom row in Fig. 7.12) and use their corresponding parameters for the initializations of the 5 populations. In 4 out of 5 runs, we find a similar result as seen for the Tetrahedron and Honeycomb patterns, meaning that only a single run provided a new pattern for the highest-scoring individual. The score of this individual was 0.240 nN which is only a small improvement from the otherwise best Random walk max drop score of 0.182 nN. However, from the other non-improving runs the initialization of the random-walk-based population provided a top score of 0.345 nN. This shows that we have better hopes of optimizing the max drop property by simply generating more configurations from these parameters than by running the genetic algorithm.

Since starting from an existing design did not give any useful results, we attempt to start from a population of random noise as well. We initialize one population with mixed porosities, having 20 individuals each for porosity $\{0.01, 0.05, 0.1, 0.2, 0.3\}$, and two populations based on a constant porosity of 0.25 and 0.5 respectively. This time the algorithm improved the highest-scoring individual throughout, but the final top scores are still not impressive. The mixed porosity start gave the highest score, being 0.299 nN. When inspecting the top five individuals in the population visually, they were all found rather similar to the starting configurations; they still looked like random noise. Thus, we do not find any significant signs that the genetic algorithm search can contribute with any higher-level pattern structures worth further investigation. We attribute this to the finding that the machine learning model is not reliable for predicting friction for configurations outside the original dataset. However, we acknowledge that the quality of the genetic algorithm results could also be affected by an inadequate number of generations in the search. Nonetheless, given our concerns regarding the machine learning model, we decided not to pursue this any further.

As an attempt to get further insight into the model predictions, we use the Grad-CAM method to examine the top-scoring individuals from the genetic algorithm search. At first, we analyzed the results for each feature map in the network, but we found that taking an average across all feature maps provided a suitable representation of the outcomes. The result for the mixed porosity search is shown in Fig. 7.15a. For comparison, we included a similar examination for the top candidates in the pattern generation search, with respect to the max drop category, for the Tetrahedron, Honeycomb and Random walk, as shown in Fig. 7.15b to 7.15d. For the mixed porosity configuration, the Grad-CAM method highlights some areas in the noise as contributing more positively than others, but we do not find any obvious structure from this. Regarding the Tetrahedron, Honeycomb, and Random Walk configurations with more organized patterns, we observe that the cuts are frequently highlighted. This finding gives some confidence to the notion that the model considers some of the relevant features in the pattern. Nonetheless, the variability of the results is too great to draw any firm conclusions. However, we notice that for certain strain values, the Grad-CAM reveals considerable “attention” toward the edge of the configuration. This especially relates to the top and bottom edge in the $\pm y$ direction. For instance, Fig. 7.15c shows a highlighting of the bottom edge in the friction prediction at a strain of 0.396 for the Honeycomb pattern. Considering that the top and bottom of the configuration are not a true edge, since these are connected to the pull blocks in the simulation, this is a bit surprising. One interpretation is that the dissipation of energy associated with the thermostat in the pull blocks might be of importance. Even though these results should be taken carefully due to the instability of the model, we note this as a topic for further investigation.

(a) Mixed porosity $p \in \{0.01, 0.05, 0.1, 0.2, 0.3\}$.

(b) Tetrahedron (1, 7, 1), ref = (1, 4).

Figure 7.15: (The figure continues on the next page)

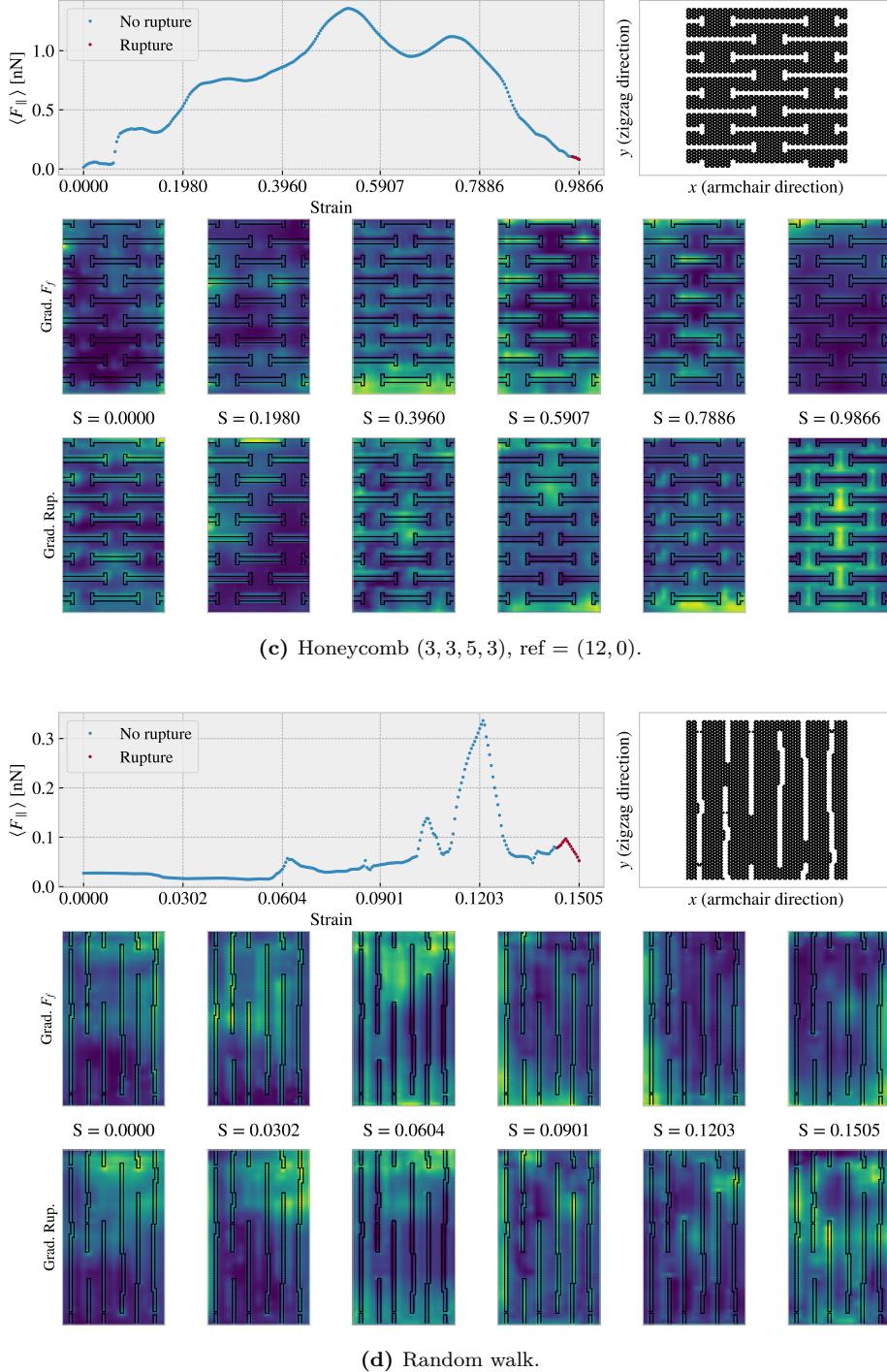
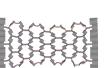


Figure 7.15: Grad-CAM analysis of selected Kirigami configurations. The top row shows the predicted friction-strain curve (left) with the rupture prediction indicated by the colors (blue: no rupture, red: rupture), and a visualization of the Kirigami graphene sheet on the hexagonal lattice (right). The remaining rows show the grad-CAM heatmaps with respect to the prediction of the friction (Grad. F_f) and rupture (Grad. rup) respectively, for various strain values corresponding to the x-axis ticks on the friction-strain curve. The edges of the cut patterns are marked with black lines for reference. The four configurations displayed are: (a) The top-scoring individual from the genetic algorithm search using a mixed porosity start. (b) The Tetrahedron (1, 7, 1) with a reference point (1, 4) which gave the highest max drop score in the investigation of pattern translation in Fig. 7.13a. (c) The Honeycomb (3, 3, 5, 3) with a reference point (12, 0) which gave the highest max drop score in the investigation of pattern translation in Fig. 7.13c. (d) The highest scoring candidate for the max drop property in the Random walk pattern generation search shown in Fig. 7.12.



Chapter 8

Negative friction coefficient

For the final part of this thesis, we concern ourselves with a proof of concept approach for the design of a device with a negative friction coefficient. From the pilot study (Chapter 6) we found the friction of the two investigated Kirigami patterns to have a non-linear relationship with strain. This is hypothesized to yield a negative friction coefficient for certain load ranges in a coupled system of load and strain. We will investigate this hypothesis further in this chapter.

8.1 Nanomachine design

We do not attempt to simulate the dynamics of any nanomachine designs. However, we propose that a coupling between load and strain could be achieved, for instance, by following a design as sketched in Fig. 8.1. Such a design could perhaps be built by rigging carbon nanotubes in a similar configuration.

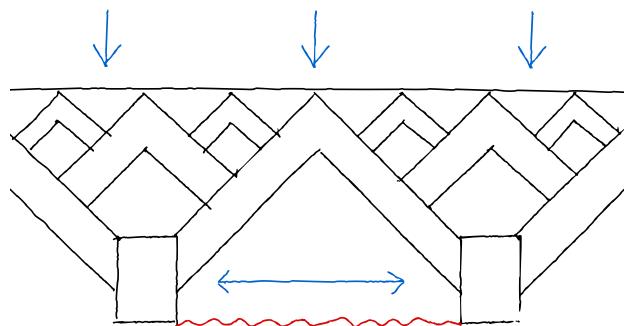
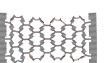


Figure 8.1: Working sketch for a nanomachine design that aims to translate applied load (from the top of the figure) to a straining of the graphene sheet (shown in red). The black boxes connected to the graphene sheet represent the pull blocks in our system.

We mimic the nanomachine coupling by implementing a load-dependent tension force to our MD simulations. So far, we have kept the pull blocks spaced by a fixed distance throughout the simulations, but now we let the pull blocks move relative to each other under the influence of tension. We let the tension force act in the $\pm y$ direction for each pull block respectively such that the force acts inwards toward the sheet. The magnitude of the tension force F_t is modeled as a linear coupling to normal load $F_t = TF_N$ by a factor T which represents the ratio for the load-to-strain coupling. We find that a ratio of $T = 6$ will provide the necessary tension for achieving a full strain range (till rupture) approximately within a load of 10 nN which was the maximum load in the pilot study. Notice that this coupling scheme is slightly different than that proposed in Eq. (6.1) where we assumed a direct coupling between load and strain $\varepsilon = RF_N$. The load-tension coupling provides a more standardized description of the proposed nanomachine since the relation between tension and strain will depend on the specific Kirigami sheet in use.

8.2 Simulating the load-tension-coupled system

For the simulation of the load-tension-coupled system, we consider the Tetrahedron (7, 5, 1) and Honeycomb (2, 2, 1, 5) patterns from the pilot study. We perform multiple simulations for different normal loads, using



100 uniformly spaced load values in the range [0, 15] nN corresponding to a tension force of [0, 90] nN. For the Tetrahedron pattern, we increase the load by a rate of 0.015 nN/ps. Due to a rapid change in the strain-tension curve for the Honeycomb pattern, we reduced the loading rate to 0.0015 nN/ps and added additional data points to the sparsely populated strain range. We compare the results to the pilot study results in Fig. 6.12 by mapping the strain to load values via an interpolation of the strain-load curve found in the coupled system. The results are shown in Fig. 8.2. Generally, we observe that the data points of the coupled system align reasonably well with the mapped data points from the pilot study. This indicates that the simultaneous loading and straining of the system does not suppress the underlying mechanism governing the non-monotonous trends. For the Tetrahedron pattern, we find an almost linear strain-tension curve. This makes for a recognizable trend in the friction-load curve similar to that seen in the friction-strain curve in Fig. 6.12. In both cases, we notice that the initial strain at the beginning of the simulation is similar to the average strain for the entire simulation. This suggests that the sliding does not contribute to a significantly increased tension in the sheet which would otherwise lead to an increased strain as well. The strain-tension curve for the Honeycomb pattern shows an interesting trend. At low tension, the strain is increasing seemingly linearly with tension. However, a drastic increase in strain from roughly 0.08 to 0.7 happens at a tension of 4.5 nN ($F_N = 0.75$ nN). Eventually, this settles off into a linear trend before reaching the rupture point. This reflects the reason for decreasing the loading rate and adding more data points to fill this gap in the strain-tension curve. The rapid increase in the strain-load curve makes for a similar rapid increase in the friction-load curve as well. Hence, we find that the Honeycomb coupled system essentially exhibits an initial discontinuous increase in friction with load, which is then followed by a longer region of decreasing friction with load. When considering the simulation frames in Appendix A, we notice that these distinct phases align rather well with the unfolding of the Honeycomb pattern. As we have previously noted, the Honeycomb pattern unfolds in segments, with one segment buckling at a time. We find that this unfolding process is initiated after passing the minimum tension limit of 4.5 nN. The increase in friction then corresponds to the unfolding phase, while the decrease in friction corresponds to the following phase of further straining. This suggests that further investigations of this mechanical transition could be beneficial for gaining a deeper understanding of the underlying mechanisms involved.

From the results in Fig. 8.2, we have found that our proposed coupled system, based on a $T = 6$ load-to-tension coupling, demonstrates a significant negative friction coefficient in certain load ranges. By considering the maximum and minimum friction points along the friction decline, we find that this corresponds to the minimal friction coefficients

$$\text{Tetrahedron: } \mu_{\min} \sim \frac{0.31 \text{ nN} - 0.90 \text{ nN}}{6.55 \text{ nN} - 4.65 \text{ nN}} = -0.31 \quad \text{Honeycomb: } \mu_{\min} \sim \frac{0.53 \text{ nN} - 1.88 \text{ nN}}{4.31 \text{ nN} - 0.71 \text{ nN}} = -0.38. \quad (8.1)$$

These results support that the use of a Kirigami sheet in a coupled system can be used to achieve a negative friction coefficient. In the pilot study Eq. (6.2) we estimated the Tetrahedron pattern to exhibit a coefficient following $-R \cdot 12.75$ nN, smaller than the Honeycomb coefficient following $-R \cdot 2.72$ nN. From the results in Eq. (8.1) we find that the resulting coefficients are more similar in value than expected from the pilot study. We attribute this to the difference in the strain-tension relationship for each pattern. However, we notice that both values can be scaled by changing the load-to-tension coupling T . If we assume that we can neglect any independent effects from load, the finding of a negative coefficient ~ -0.3 using $T = 6$ suggests that we have the relation

$$\mu_{\min} \sim -\frac{0.3}{6}T = -0.05T.$$

The load-to-tension coupling T is purely governed by the assumptions for the nanomachine. Thus, we note that even more (or less) extreme values might be accessible through a realization of the nanomachine design.

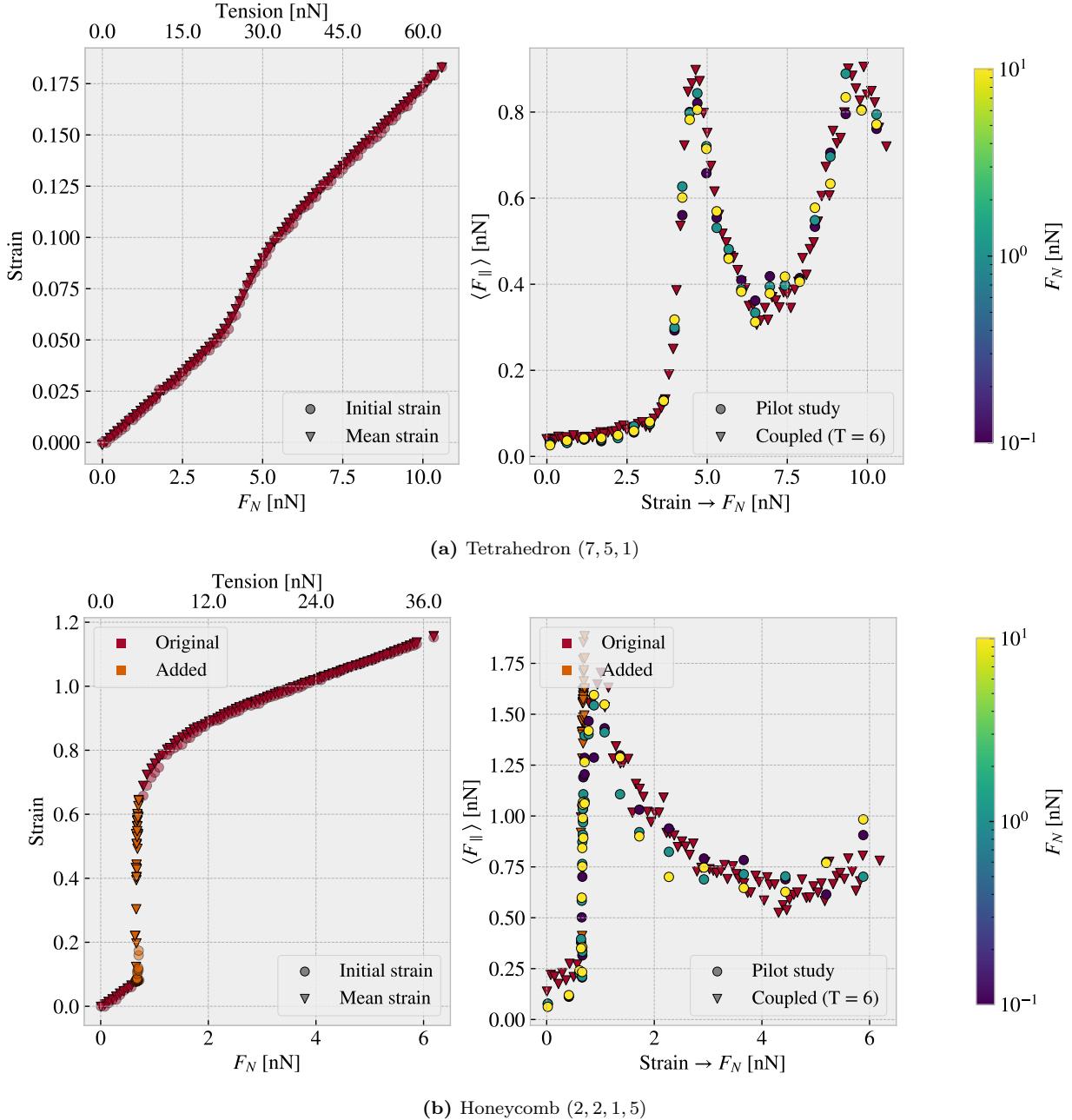
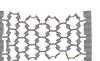


Figure 8.2: Investigation of the friction response for a coupled system using (a) the Tetrahedron (7, 5, 1) and (b) the Honeycomb (2, 2, 1, 5) pattern from the pilot study. The pull blocks are allowed to move relative to each other under the influence of a tension force F_t modeled as $F_t = T F_N$ for normal load F_N and a load-to-tension ratio $T = 6$. The right panel shows the friction-load curve for the coupled system (orange and red triangles) in comparison to the results from the pilot study where the pull blocks are locked at a fixed separation distance and with a constant normal load (Fig. 6.12). The pilot study results have their strain values mapped to the normal load axis through the observed strain-load curve (left panel) for the coupled system. Their original normal loads are indicated via the colors in the colorbar. The left panel shows the strain-tension curve. The circles denote the initial strain, at the beginning of the simulation, while the triangles denote the mean strain for the whole simulation. For the Honeycomb pattern, we have added more data points in the region where the strain-tension curve changed rapidly.



Chapter 9

Summary

In this thesis, we have studied the nanoscale friction of a Kirigami graphene sheet under the influence of strain using molecular dynamics (MD) simulations. We have developed a numerical tool for generating diverse Kirigami designs which we have utilized to create a dataset for the frictional behavior depending on Kirigami pattern, strain, and load. Our findings suggest that the frictional behavior of a Kirigami sheet is highly dependent on the geometry of the pattern and the strain conditions. We have observed that the out-of-plane buckling can be associated with a non-monotonous friction-strain relationship which can be utilized to demonstrate a negative friction coefficient in a system with coupled load and strain. Moreover, we have investigated the possibility to use machine learning on this dataset and we have attempted an accelerated search for the optimization of various friction properties. Our findings imply that machine learning can be feasible for this approach, but additional data is required to establish a more reliable foundation for the prediction on new Kirigami patterns. In this chapter, we will provide a summary of our findings and draw conclusions based on the results obtained. Finally, we will suggest some topics for further research.

9.1 Summary and conclusions

9.1.1 Designing an MD simulation

We have designed an MD simulation for the examination of friction for a graphene sheet sliding on a silicon substrate. The key system features were the introduction of the pull blocks, defined as the end regions of the sheet with respect to the sliding direction, which was utilized for applying normal load and sliding the sheet. The pull blocks were made partly rigid and used to employ a thermostat as well. Through an analysis of the friction forces retrieved from the sliding simulations, we have established a standardized metric for kinetic friction. In particular, we measured the force exerted by the substrate on the full sheet, including the pull blocks, with respect to the sliding direction. We determined the kinetic friction as the mean value of the friction force for the last half of the simulation. The uncertainties were estimated based on the fluctuations in the running mean. We found that the assessment of static friction was ambiguous for our simulation and did not pursue this further. From the analysis of the force traces, friction force vs. time, we identified the friction behavior in our default parameter simulations as being in the smooth sliding regime. This is attributed to the choice of a relatively high sliding speed ($v = 20 \text{ m/s}$) and infinitely stiff springs ($K = \infty$) for the tethering of the sheet. This was supported by a demonstration of a transition to the stick-slip regime through the use of softer springs ($K = 10 \frac{\text{N}}{\text{m}}$) and a decrease in sliding speed ($v = 1 \frac{\text{m}}{\text{s}}$). By conducting a more systematic investigation of the effects of temperature, sliding speed, spring constant and timestep, we identified a set of default values based on numerical stability and computational cost. During this process, we aimed to select the variables that would maintain relatively stable measures for friction with moderate perturbations around these default values. We found that friction increased with temperature which is in disagreement with the Prandtl–Tomlinson model and most experimental results. However, this agrees with the predictions from the 2D Frenkel–Kontorova model and other MD studies which attribute this to ballistic sliding. In the absence of clear indications regarding an appropriate temperature choice, we opted for the standard choice of room temperature ($T = 300 \text{ K}$). Furthermore, we found friction to increase with sliding speed as expected from both experimental and numerical studies. We found an approximately viscous friction $F_{\text{fric}} \propto v$ as predicted by the Prandtl–Tomlinson model, with some signs of phonon resonance at certain sliding speeds which aligns with the predictions from the Frenkel–Kontorova models. We chose a rather high sliding speed of 20 m/s mainly for the consideration of computational costs. For the spring constant, we found decreasing friction with increasing stiffness of the springs. This is associated with the transition from a stick-slip-influenced



regime toward smooth sliding and can be attributed to an underlying change in commensurability. This is predicted by the Frenkel-Kontorova models and supported by both numerical and experimental results. The choice of an infinitely stiff spring was made from an assessment of the variation in friction with perturbations in the spring constant value. Finally, we confirmed that a timestep of 1 fs provides reasonable numerical stability. However, based on fluctuations with timestep we find that the uncertainty in the simulations might be higher than first estimated. For the non-strained Kirigami sheet, these fluctuations were on the order of $\pm 0.017 \text{ nN}$ for the evaluation of the kinetic friction.

9.1.2 Generating Kirigami patterns

In order to investigate the effects of Kirigami design we have created a numerical tool for generating various patterns. By defining an indexing system for the hexagonal lattice structure we were able to define the Kirigami cut patterns as a 2D binary matrix for numerical implementation. We have selected two macroscale designs, which we denote the *Tetrahedron* and *Honeycomb* patterns, based on their ability to exhibit out-of-plane buckling when subjected to strain. By digitalizing the designs for the hexagonal graphene lattice, we found that the characteristic design features can be translated to the nanoscale, as we observed similar out-of-plane buckling in MD simulations. Through our numerical tool, we were able to create an ensemble of perturbed variations of these patterns which yielded approximately 1.35×10^5 and 2.025×10^6 unique configurations for the Tetrahedron and Honeycomb patterns respectively. When considering the possibility to translate the periodic patterns on the sheet, the number of unique configurations can be increased by approximately a factor of 100. To introduce some random design features, we have developed a tool for generating Kirigami patterns based on random walks. The tool includes mechanisms such as bias, avoidance of existing cuts, preference for maintaining a direction, and procedures for repairing the sheet for simulation purposes. In general, we found that the numerical tools for creating Kirigami designs surpassed our computational capabilities for conducting MD simulations with various loads and strains for each of the designs. Our MD-based dataset only utilized a subset of configurations with 9660 data points based on 216 Kirigami configurations (Tetrahedron: 68, Honeycomb: 45, Random walk: 100, Pilot study: 3). Hence, we argue that the Kirigami generative tool can be valuable for further studies on an extended dataset.

9.1.3 Friction control using Kirigami design and strain

We have investigated the frictional behavior of the Tetrahedron and Honeycomb patterns in comparison to a non-cut sheet under various strains and loads. Initially, we observed that straining the Kirigami sheets in a vacuum resulted in an out-of-plane buckling. When adding the substrate to the simulation, this translated into a decreasing contact area with strain. We found the Honeycomb sheet to exhibit the most significant buckling with a corresponding reduction of relative contact to approximately 43%, whereas the non-cut sheet did not produce any significant buckling in comparison. For the Kirigami sheets, we found that friction initially increased with strain, which made for increasing friction with decreasing contact area. As the strain continued to increase the friction-strain curve exhibited highly non-linear trends with strong negative slopes as well (see Fig. 6.12). During the full strain, the contact area was decreasing monotonically except for a slight increase just before rupturing. These results contradict the asperity theory hypothesis of decreasing friction with decreasing contact area, which is also supported by the predictions of the 2D Frenkel-Kontorova model suggesting increasing friction with (contacting) system size. Thus, we conclude that the contact area is not a governing mechanism for the friction-strain relationship observed. From the investigation of the friction-load relationship, we found that both friction and contact area increased slightly with load, but we were not able to find significant evidence of any linear relationship between friction and contact area as predicted by asperity theory. The non-cut sheet did not show any dependency on the strain as both the friction and the contact area remained constant with strain. Thus our findings suggest that a changing contact area and the strain-induced friction effects might be associated with an underlying mechanism related to the buckling of the sheet. When analyzing the independent effect of the non-strained Kirigami sheets, we found a slight increase in friction between the non-cut sheet and the Kirigami sheets. However, this increase was one order of magnitude lower than the friction changes induced by the strain in combination. Therefore, we can conclude that the observed friction behavior cannot be attributed solely to the effects of the non-strained Kirigami sheet or the tension induced by the strain in a non-cut sheet. When considering the friction dependency with load, we generally found a weak dependency corresponding to a friction coefficient on the order of 10^{-4} – 10^{-5} even though we could not confirm any clear relationship. This is best attributed to the superlubric state of the graphene sheet as seen in other studies as well. The slope of the friction-load curve was not considerably affected by the straining of the Kirigami sheet which led us to the conclusion that strain-induced effects are dominant in comparison to any load-related effects.

When considering our findings in light of previous related results, we find it plausible that the governing mechanism for the observed friction effects is related to commensurability as predicted by the Frenkel-Kontorova models. Since we have observed extremely low friction in our system, we argue that the non-deformed sheet corresponds to an incommensurable configuration. This is supported by the fact that we were not able to lower the friction below the original starting point. Additionally, this also aligns with the observation that the introduction of the non-strained Kirigami designs increased friction slightly. Since the Kirigami designs correspond to the removal of atoms from the sheet, this can be hypothesized to relax the incommensurability to some extent. When the Kirigami sheet buckles during stretching, it allows for a considerable rearrangement of the atoms in contact with the substrate. Hence, it may transition in and out of commensurable configurations, which could explain the non-monotonous trend for the friction-strain curve. One way to test this hypothesis is to alter the simulation conditions such that the non-strained sheet starts in a commensurable phase. This might be achieved by a softening of the springs for tethering and a lowering of the sliding speed since this was found to yield stick-slip behavior which can be associated with a commensurable phase. Another way is to reorient the sheet or change the sliding direction as reported in both numerical and experimental results. Then we might find a transition from a commensurable to an incommensurable case during the initial straining which would result in a lowering in friction with respect to the starting point.

9.1.4 Capturing trends with machine learning

By utilizing the numerical tool for generating Kirigami designs, we have created a MD-based dataset for the frictional behavior depending on Kirigami design, load and strain. The dataset reveals some general correlations with mean friction, such as a positive correlation to strain (0.77) and porosity (0.60), and a negative correlation to contact area (-0.67). These results align with the findings in the pilot study, suggesting that the change in friction is associated with cuts in the sheet (porosity) and a changing contract area representing out-of-plane buckling.

By defining the friction property metrics: $\min F_{\text{fric}}$, $\max F_{\text{fric}}$, $\max \Delta F_{\text{fric}}$ and $\max \text{drop}$ (maximum decrease in friction with strain), we investigated the top design candidates within our dataset. From these results, we found no indication of the possibility to reduce friction with the Kirigami approach since the non-cut sheet provided the overall lowest friction. Furthermore, among the top candidates, we found that a flat friction-strain profile is mainly associated with little decrease in the contact area and vice versa. These observations are consistent with the results of the pilot study and support the hypothesis that commensurability plays a key role in governing the behavior of the system. In terms of the maximum properties, we observed an improvement compared to the values obtained in the pilot study, with the Honeycomb patterns exhibiting the highest scores. This indicates that the dataset contains relevant information for optimizing these properties since it includes examples of design improvements.

For the machine learning investigation, we have implemented a VGGNet-inspired convolutional neural network with a deep “stairlike” architecture: C32-C64-C128-C256-C512-C1024-D1024-D512-D256-D128-D64-D32, for convolutional layers C with the number denoting channels and fully connected (dense) layers D with the number denoting nodes. The final model contained 1.3×10^7 parameters and was trained using the ADAM optimizer for a cyclic learning rate and momentum scheme. We trained the network for a total of 1000 epochs while saving the best model during training based on the validation score. The model validation performance gives a mean friction R^2 score of $\sim 98\%$ and a rupture accuracy of $\sim 96\%$. However, we got lower scores for a selected subset of the Tetrahedon ($R^2 \sim 88.7\%$) and Honeycomb ($R^2 \sim 96.6\%$) patterns based on the top 10 max drop property scores respectively. The scores obtained were lower, even though the selected configurations were partly included in the training data and the hyperparameter selection favored the performance on this selected set. Thus we conclude that these selected configurations, associated with a highly non-linear friction-strain curve, represent a bigger challenge for machine learning modeling. One interpretation is that these involve the most complex dynamics and perhaps that this is not readily distinguished from the behavior of the other configurations which constitute the majority of the dataset. By evaluating the ability of the model to rank the dataset based on property scores, we found that it was able to effectively represent the top three scores for the maximum categories. However, the ranking for the minimum friction property was lacking, which we attribute to the requirement of higher prediction precision that the model did not meet. To obtain a more accurate evaluation of the model’s performance, we generated a test set using MD simulations for some of the random-walk-based suggestions obtained from the accelerated search. The results showed a significantly worse performance compared to the validation set, with a two-order of magnitude higher loss and a negative R^2 score for the mean friction property. The negative R^2 score suggests that the model’s predictions were worse than simply predicting the mean value of the true data. However, by reevaluating the hypertuning and choosing solely based on validation loss,



we still found poor results on the test set. This suggests, that the inadequate performance is not solely due to a biased hypertuning process, but rather because our original dataset did not cover a sufficiently diverse range of Kirigami configurations. The validation scores indicate that the use of machine learning is a feasible approach for addressing this problem, as we were able to identify some general trends in the data. Nonetheless, from the test scores, it is evident that further improvements to the dataset are necessary in order to develop a reliable model.

9.1.5 Accelerated search for Kirigami patterns

Using the machine learning model we performed two types of accelerated search. One by evaluating the property scores of an extended dataset and another with the use of the genetic algorithm approach. For the extended dataset search, we used the developed pattern generators to generate 1.35×10^6 Tetrahedron, 2.025×10^7 Honeycomb and 10^4 Random walk patterns. The search results for the minimum friction property indicate a preference for low cut density. This aligns with the overall observation that the dataset does not provide any suggestions for a further reduction in friction. The search for the maximum properties resulted in some minor score increases, but the suggested candidates were mainly overlapping with the original dataset. By investigating the sensitivity to translations of the Tetrahedron and Honeycomb patterns, we observed significant variations in the model predictions with only minor translations. This might be attributed to a physical dependency since these translations affect the edge of the sheet. However, given the poor model performance on the test set, we believe it is more likely that this variation is due to an insufficiency in the model caused by the limitations of the dataset.

In our investigation of the genetic algorithm approach, we used a starting population that was based on the results from the extended dataset accelerated search, as well as some randomly generated initializations with different porosity values. However, this approach did not provide any noteworthy indication for new design structures worth more investigation. In general, the initialization of the population itself proved to be a more promising strategy than the genetic algorithm. We acknowledge that further effort could potentially yield useful results with the genetic algorithm approach. However, we believe that the current lack of promising results can be attributed to the uncertainty of the model which where the reason for not pursuing this any further.

By considering the Grad-CAM explanation method, we observed that the model predictions were often substantially reliant on the top and bottom edges of the Kirigami configurations. This was unexpected since these edges are not true edges but are connected to the pull blocks used in the simulation. Despite the model's uncertain predictions, we speculate that this may be due to the thermostat effects from the pull blocks. Therefore, we note this as a feature worth investigating in the simulations.

9.1.6 Negative friction coefficient

Based on our initial investigations of the Kirigami sheet, we have discovered a highly non-linear friction-strain relationship. By proposing a linear coupling between load and strain with ratio R , we found that these results suggest the possibility to utilize the negative slope on the friction-strain curve to achieve a negative friction coefficient. Based on the local decrease in friction, using the Tetrahedron (7, 5, 1) and Honeycomb (2, 2, 1, 5) patterns from the pilot study, we estimate that the average coefficient within this range will be $-R \cdot 12.75 \text{ nN}$ for the Tetrahedron pattern and $-R \cdot 2.72 \text{ nN}$ for the Honeycomb pattern.

To investigate this hypothesis, we conducted a simulation with a coupling between load and sheet tension, mimicking a nanomachine attached to the sheet, using the Kirigami configurations from the Pilot study. We observed that the non-linear behavior in the friction-strain curve also translated into a non-linear friction-load relationship for the coupled system. Additionally, we found that the Honeycomb pattern exhibited a non-linear strain-tension curve which resulted in an almost discontinuous increase in friction for the initial increase in load. We attribute this feature to an unfolding process visually confirmed from the simulation frames. For the coupled system with a load-to-tension ratio of $T = 6$, we found regions in the friction-load curve with significant negative slopes. By considering the maximum and minimum points for these regions we estimated the average friction coefficient to be -0.31 in the load range $F_N = [4.65, 6.55] \text{ nN}$ for the Tetrahedron pattern and -0.38 in the range $F_N = [0.71, 4.31] \text{ nN}$ for the Honeycomb pattern. These results can be scaled by adjusting the load-to-tension ratio T . By neglecting any independent effects from load, these results suggest that a negative friction coefficient on the order of ~ -0.3 corresponds to the general coefficient $-0.05T$.

Based on our investigations, we have found that the combination of Kirigami cuts and strain has significant potential for controlling friction. Specifically, we have demonstrated that by enforcing a coupling between load and strain through tension, it is possible to achieve a negative friction coefficient. Therefore, we believe that this approach could be promising for developing novel friction-control strategies.

9.2 Outlook

In this thesis, we have demonstrated that certain Kirigami designs exhibit non-linear friction behavior with strain. This discovery was made through an exploration of different designs, which invites further investigation into the underlying mechanisms of this phenomenon. To this end, it would be valuable to choose only one or two selected designs, such as the Tetrahedron and Honeycomb patterns, and perform an extended study of the effects of various physical parameters on the friction-strain curve.

First of all, we suggest an investigation of how the friction-strain curve depends on temperature, sliding speed, spring constant, and loads for an increased range $F_N > 100nN$. This is especially interesting in the context of physical conditions leading to a stick-slip behavior since our study takes a basis in smooth sliding friction. Moreover, it would be valuable to verify that the choices for relaxation time, pauses, interatomic potentials and substrate material are not critical for the qualitative observations found for the friction-strain relationship. Especially, the Adaptive Intermolecular Reactive Empirical Bond Order (AIREBO) potential for the modeling of the graphene sheet might be of interest. In this context, it might also be useful to investigate the effects of excluding adhesion from the simulations.

In order to investigate the hypothesis of commensurability as a governing mechanism, we suggest an investigation of the friction-strain curve for different scan angles. If commensurability is an important factor, we hypothesize that the friction-strain curve will exhibit different qualitative shapes at varying scan angles. Additionally, it may be interesting to investigate the friction-strain relationship under a uniform load on the entire sheet to gain insight into how the loading distribution affects the out-of-plane buckling and associated commensurability effect.

Another topic worth exploring is the impact of scale and edge effects. This includes an investigation of scaling the ratio of the sheet area to the sheet edge length. However, the machine learning predictions also suggest that it may be valuable to study Kirigami-induced edge effects as we translate the patterns on the sheet. With this regard, we would also suggest a more detailed study of the effect of the thermostat in the pull blocks which is suggested to have a possible impact when judging from the Grad-CAM analysis.

Regarding the machine learning approach, our findings indicate that there is a significant need to expand the dataset. In order to get more insight into this issue one could use unsupervised clustering techniques like the t-Distributed Stochastic Neighbor Embedding (t-SNE) to visualize the distribution of Kirigami configurations in the dataset. Another valuable approach is the active learning method similar to that used by Hanakata et al. [6]. That is, we extend the dataset using the top candidates of a machine learning-driven accelerated search and repeat the process of training the model, searching for new candidates and extending the dataset. This provides a direction for the extension of the dataset which could lead to a more efficient approach to address the dataset problem. We note that one can also create a dataset based on a fixed Kirigami design and vary the physical parameters to support the investigations mentioned above. For both variations, we believe that the results could benefit from a consideration of more advanced model architectures and machine learning techniques. For instance, we suggest increasing the receptive field in the convolutional part of the model, by the use of larger strides or dilated convolution. If we can develop a reliable machine learning model, it would invite further studies of inverse design methods such as GAN or diffusion models.



Appendices



Appendix A

Strain simulation frames

Fig. A.1 and Fig. A.2 show the sheet-substrate system during a strain simulation. The used Kirigami patterns are the Tetrahedron (7, 5, 1) and Honeycomb (2, 2, 1, 5) respectively which are used in the pilot study Chapter 6. The simulations correspond to the investigation of contact area during strain as seen in Fig. 6.9. We strain the sheet until rupture and use a zero normal load. The remaining parameters follow the default values from Table 6.1. Notice also that the flip book animation in the page corners corresponds to the Honeycomb simulation in Fig. A.2. The side view is shown in the left page corners while the top view is shown in the right page corners. Each page represents a frame in the strain simulation although it reverses and plays backward from the middle of the thesis. By going through the pages quickly one can get a feeling for the motion in the simulation. This works better in print by releasing the pages held with the thumb.



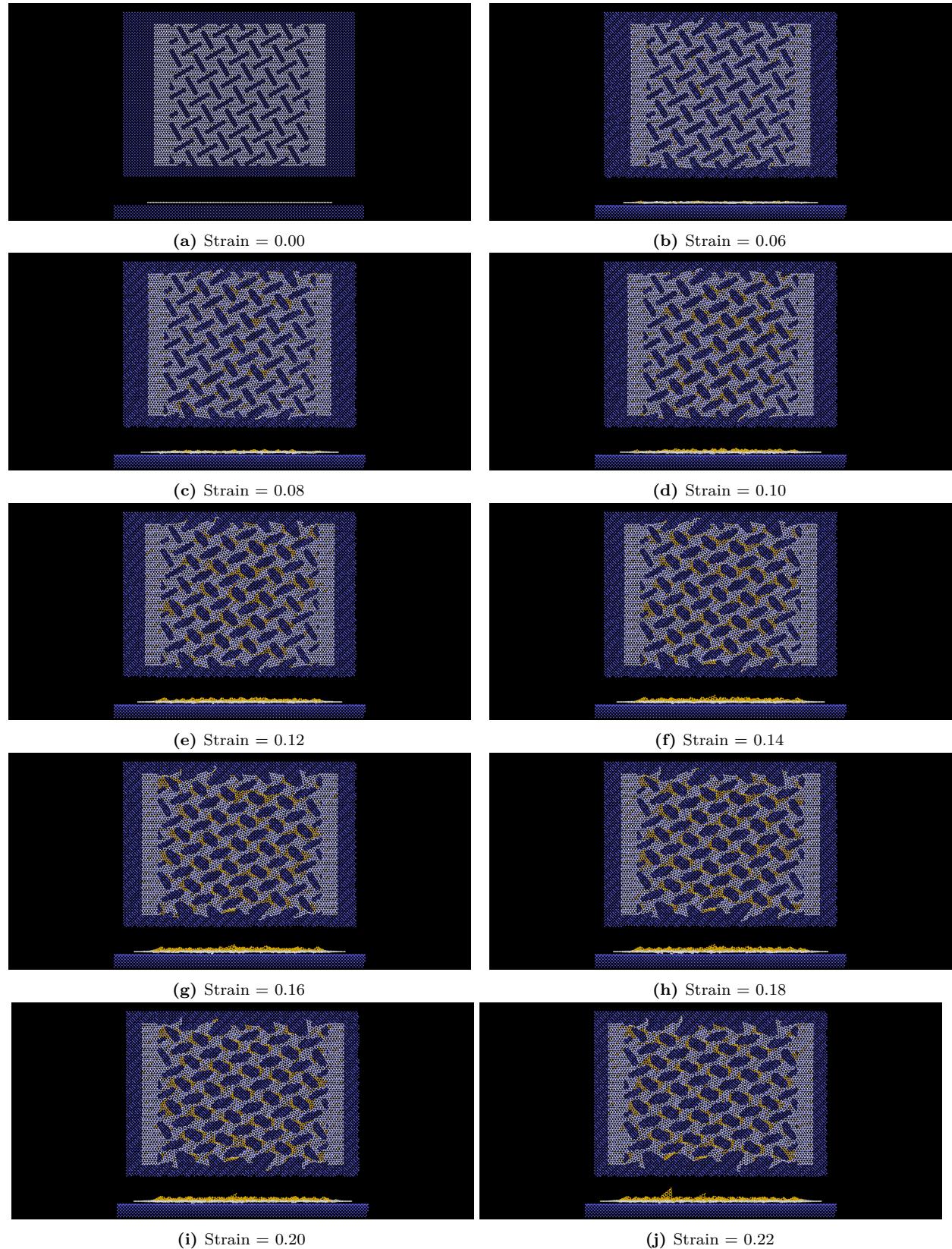


Figure A.1: Straining of the Tetrahedron (7,5,1) pattern against the substrate. The top part of each frame (a) to (j) shows a top-down view into the x-y plane, with the y-direction on the horizontal axis and the x-direction on the vertical axis. The bottom part of each frame shows a side view of the system, with the y-direction on the horizontal axis and the z-direction on the vertical axis. White-colored atoms indicate graphene sheet atoms in contact with the substrate while the yellow-colored atoms are not in contact. The substrate is shown in blue.

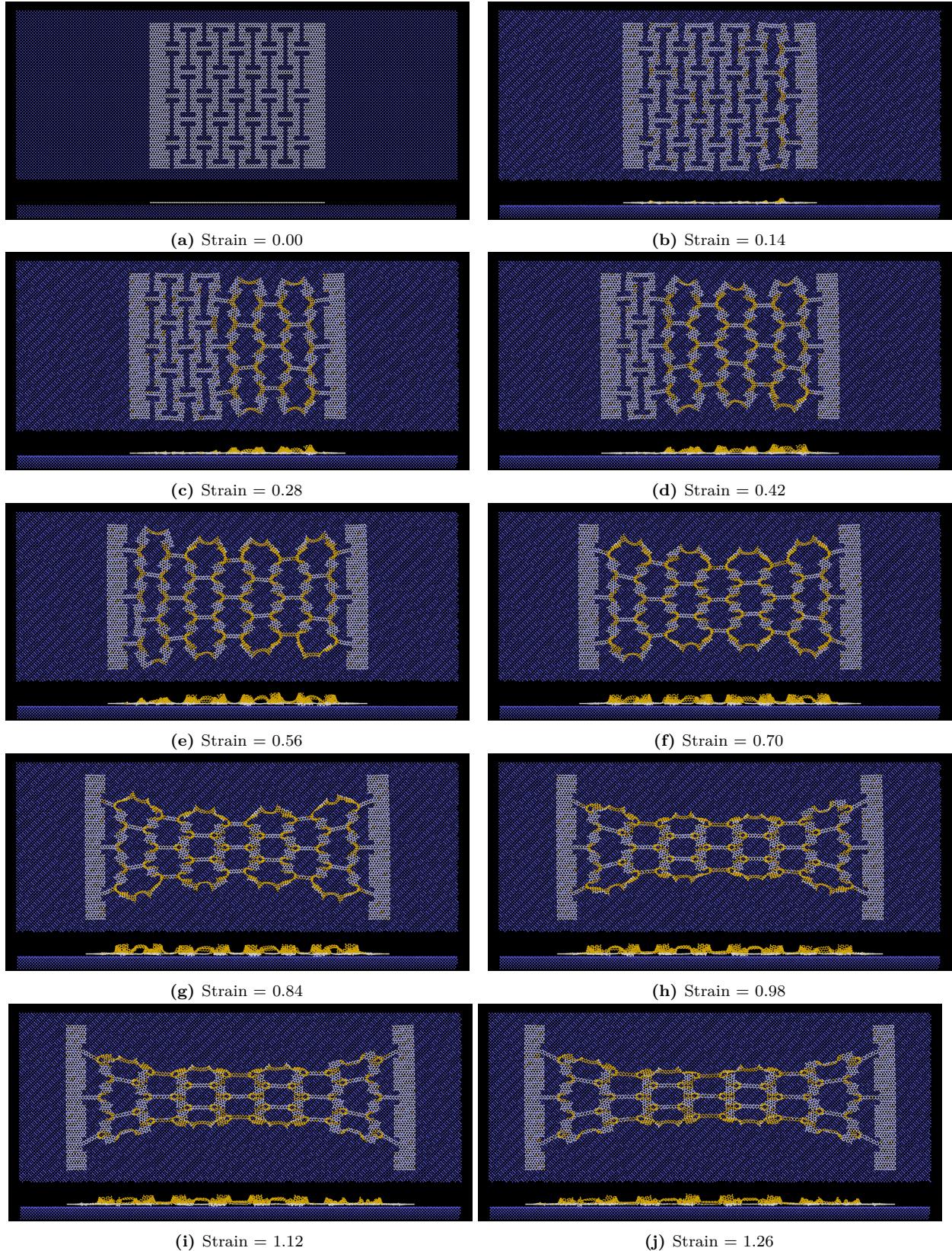


Figure A.2: Straining of the Honeycomb (2,2,1,5) pattern against the substrate. The top part of each frame (a) to (j) shows a top-down view into the x-y plane, with the y-direction on the horizontal axis and the x-direction on the vertical axis. The bottom part of each frame shows a side view of the system, with the y-direction on the horizontal axis and the z-direction on the vertical axis. White-colored atoms indicate graphene sheet atoms in contact with the substrate while the yellow-colored atoms are not in contact. The substrate is shown in blue.



Appendix B

Dataset Kirigami patterns

Visualization of the dataset configurations corresponding to the Tetrahedron, Honeycomb and Random walk classes respectively. Note that the three configurations used in the pilot study: Non-cut, Tetrahedron (7, 5, 1) and Honeycomb (2, 2, 1, 5), are not explicitly shown here since the first is trivial (seen in Fig. 5.2) and the latter two are also part of the dataset shown here, but with a random reference. Thus we show the remaining 213 configurations organized as

1. 68 Tetrahedron configurations shown in Fig. B.1 to B.3.
2. 45 Honeycomb configurations shown in figure Fig. B.4 to B.5.
3. 100 Random walk configurations shown in figure Fig. B.6 to B.9 and supplementary details are given in Table B.1.

Notice that the pull blocks are not shown in these visualizations.



B.1 Tetrahedron

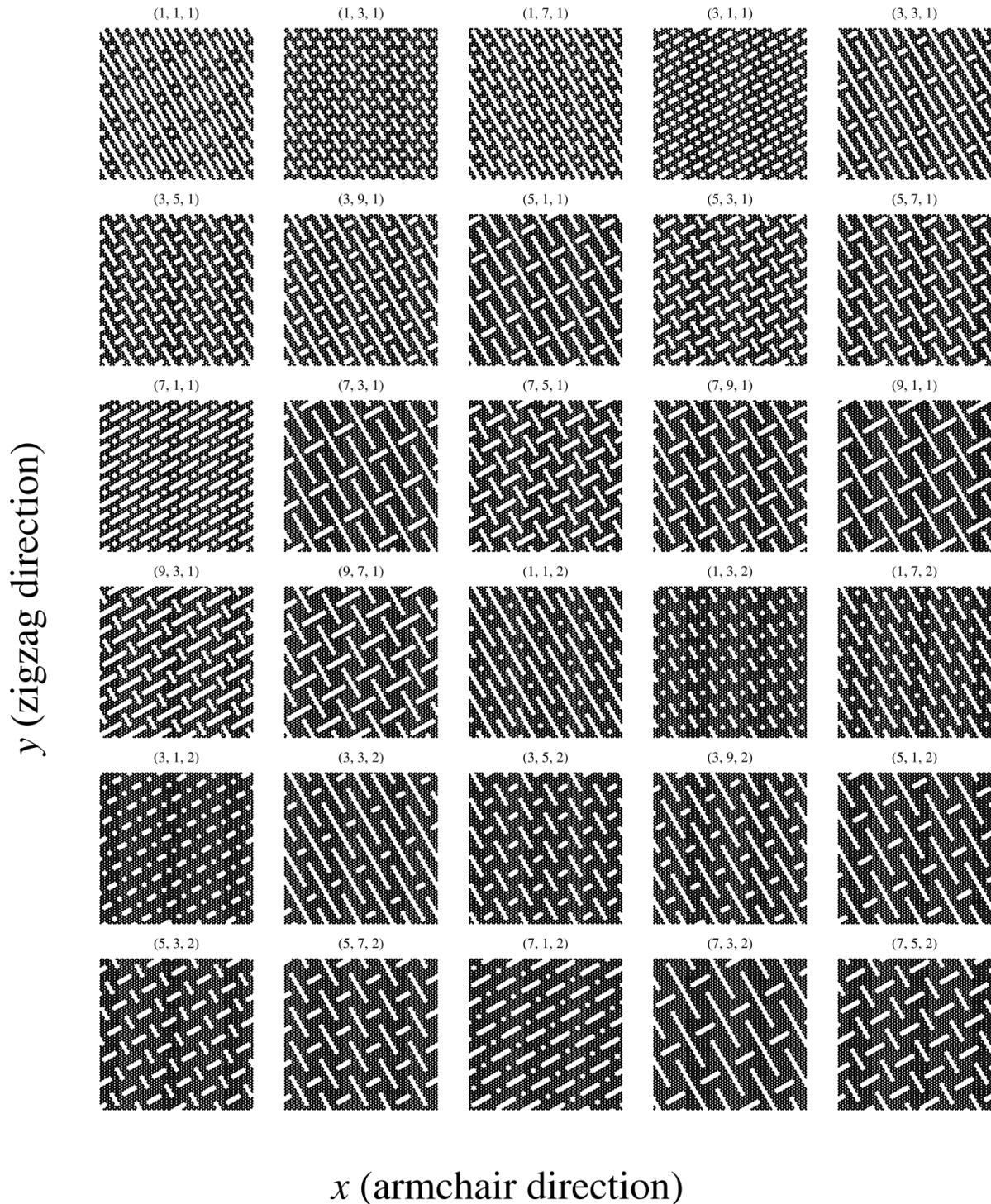


Figure B.1: Tetrahedron patterns (0–29). Grey circles depict atoms in the graphene sheet.

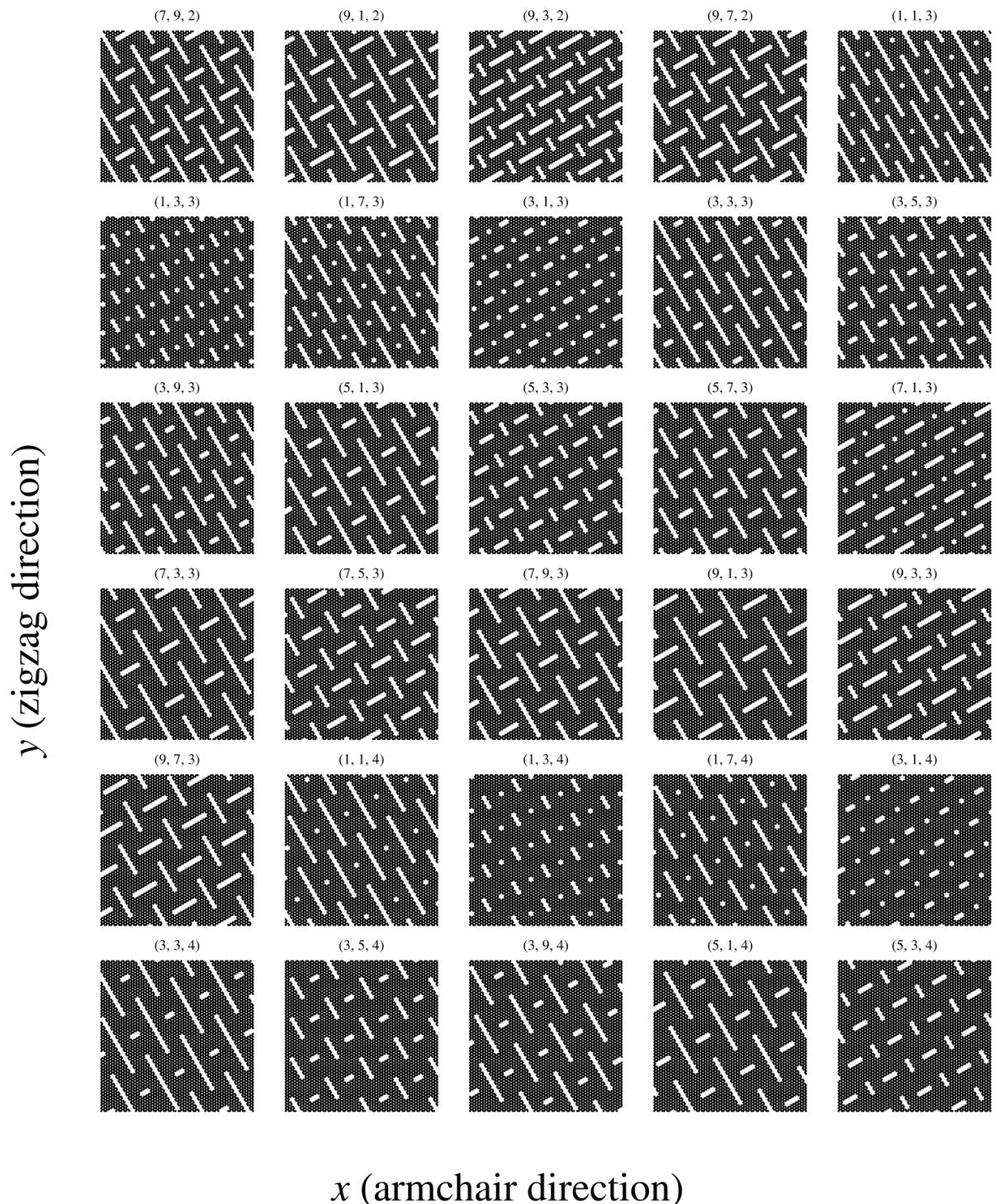


Figure B.2: Tetrahedron patterns (30–59). Grey circles depict atoms in the graphene sheet.

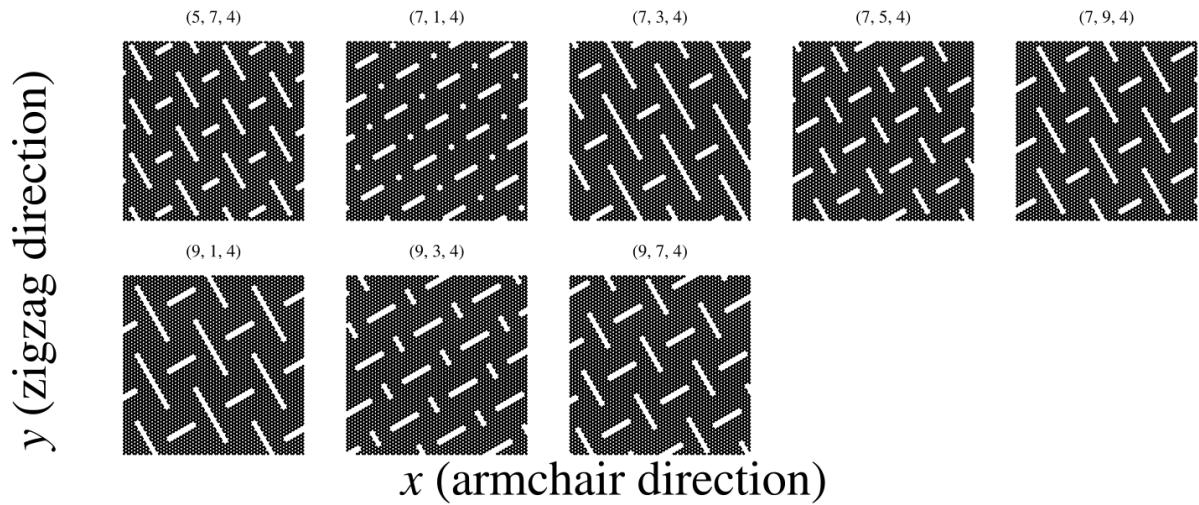


Figure B.3: Tetrahedron patterns (60–67). Grey circles depict atoms in the graphene sheet.

B.2 Honeycomb

y (zigzag direction)

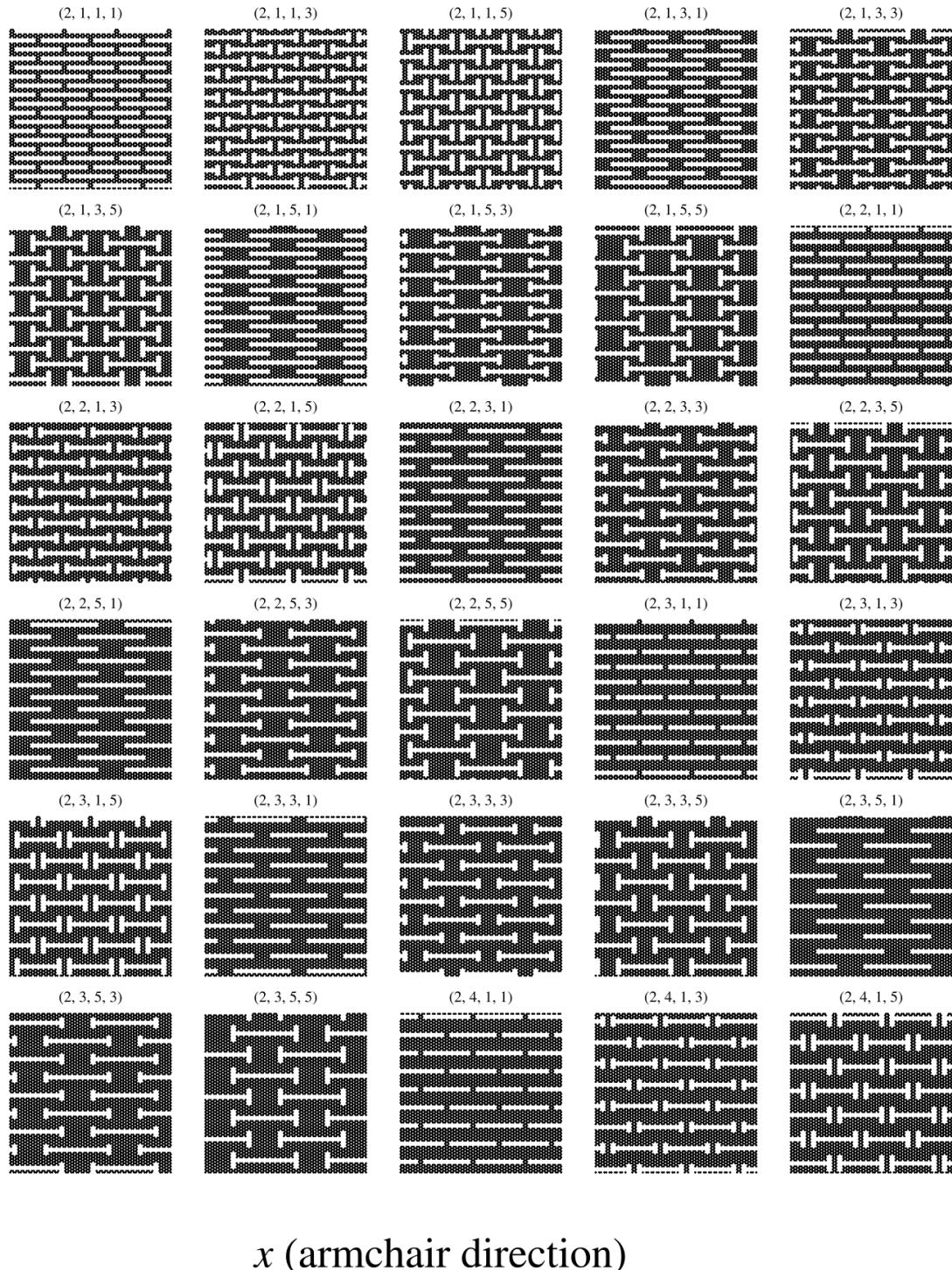


Figure B.4: Honeycomb patterns (0–29). Grey circles depict atoms in the graphene sheet.

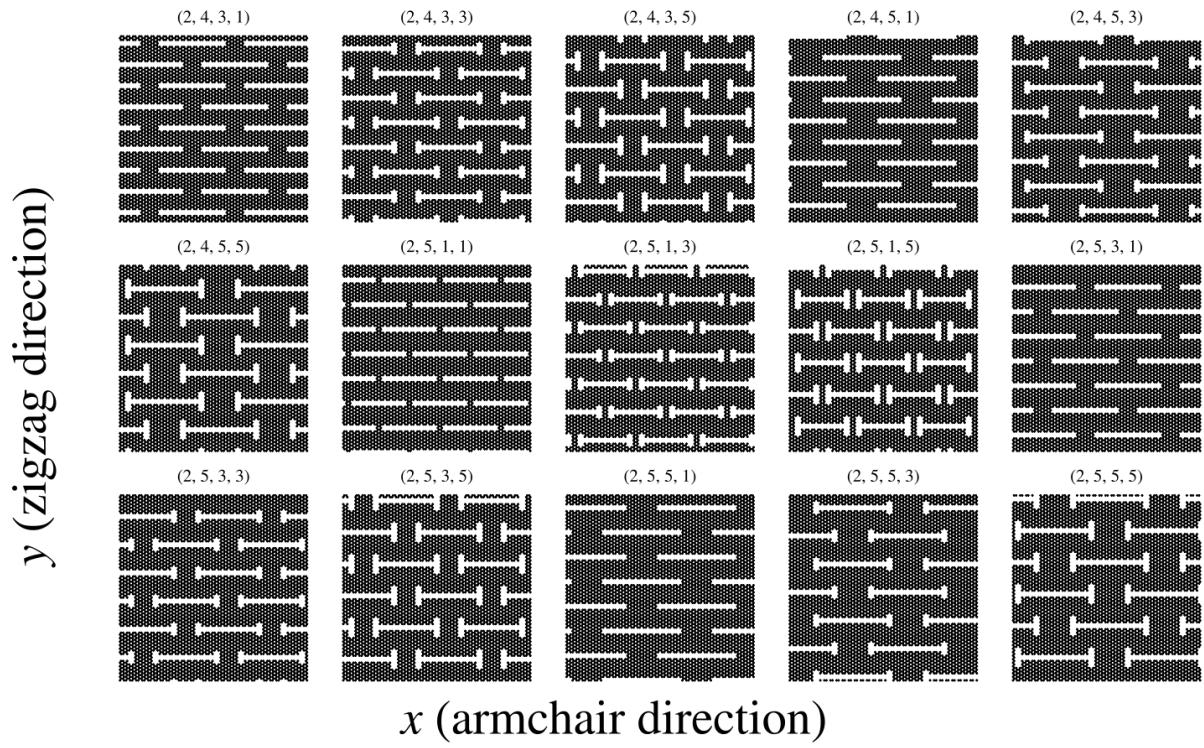


Figure B.5: Honeycomb patterns (30–44). Grey circles depict atoms in the graphene sheet.

B.3 Random walk

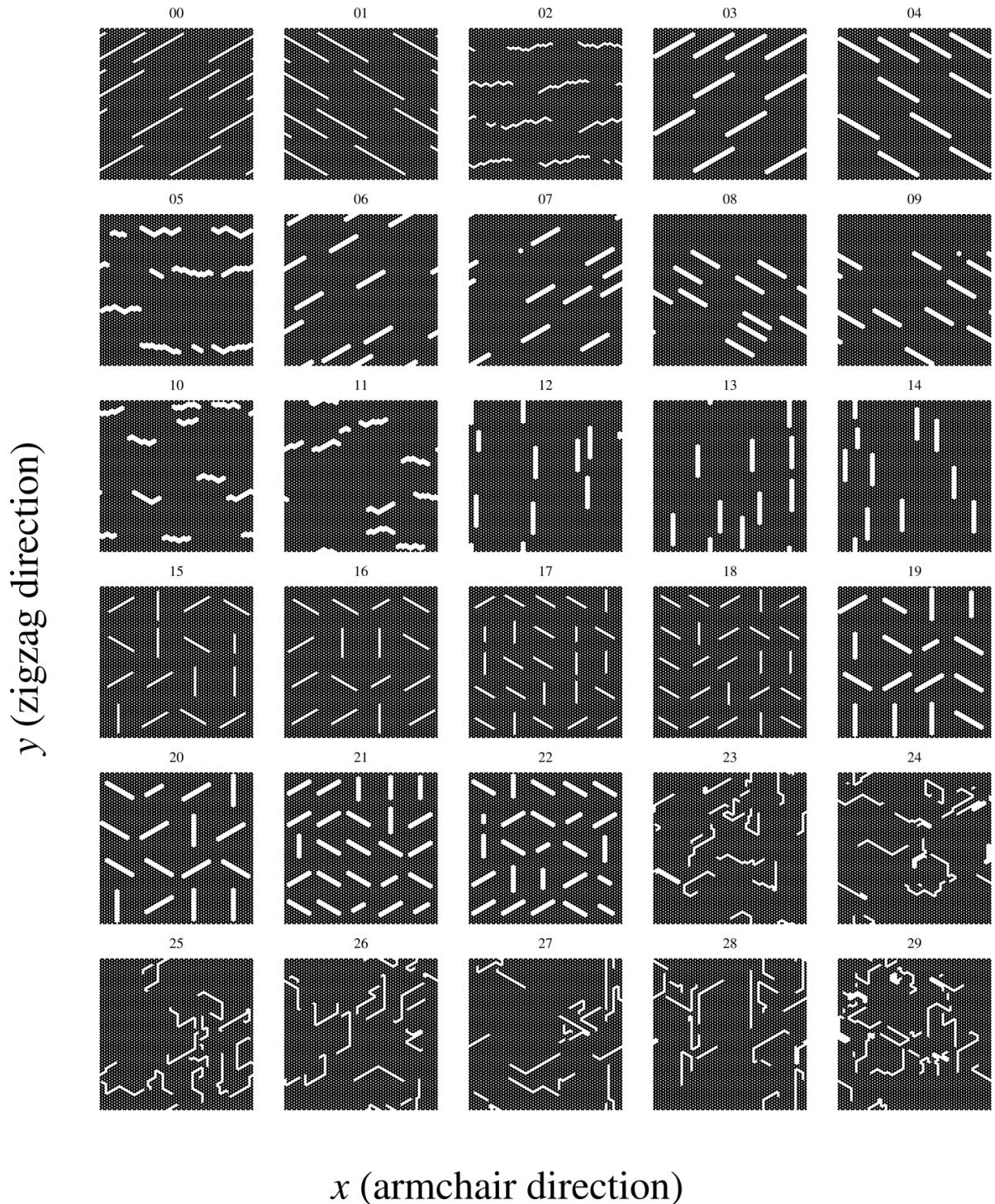


Figure B.6: Random walk patterns (0–29). Grey circles depict atoms in the graphene sheet.



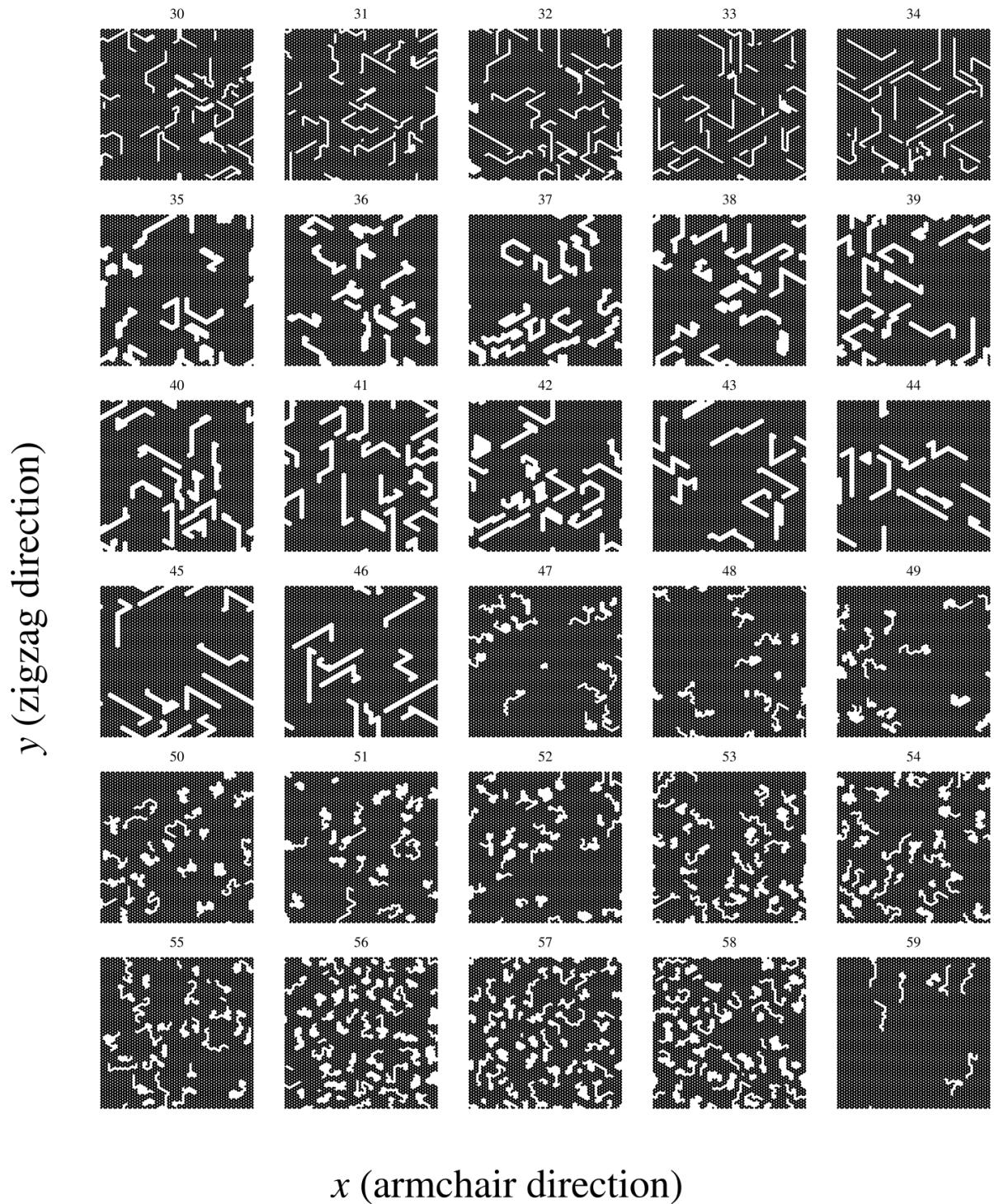


Figure B.7: Random walk patterns (30–59). Grey circles depict atoms in the graphene sheet.

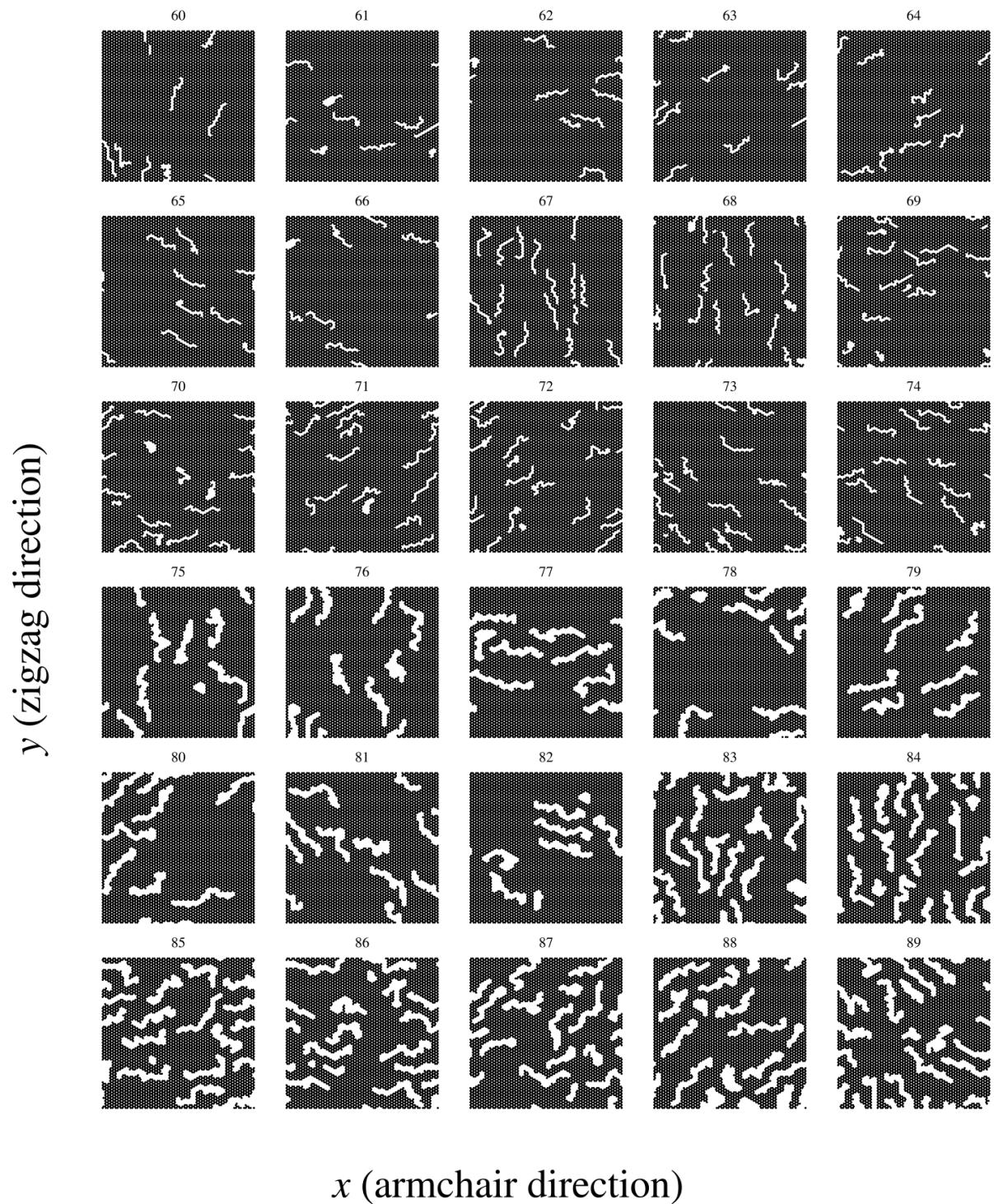


Figure B.8: Random walk patterns (60–89). Grey circles depict atoms in the graphene sheet.



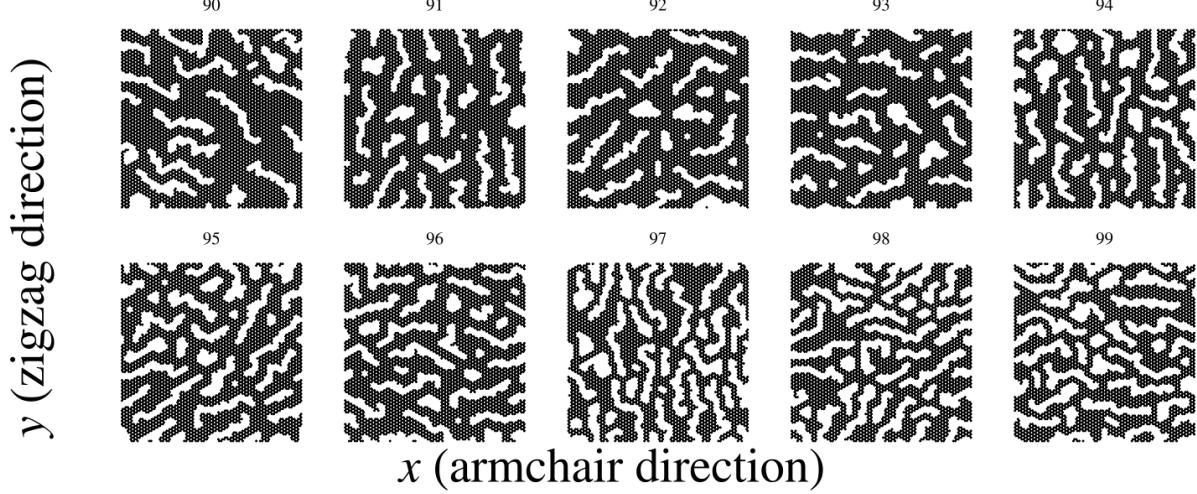


Figure B.9: Random walk patterns (90–99). Grey circles depict atoms in the graphene sheet.

| Index | Num. walks | Max. steps | Min. dis. | Bias (dir., temp.) | Connec-tion | Avoid invalid | RN6 | Grid start | Cen-tering | Stay or break |
|-------|------------|------------|-----------|--------------------|-------------|---------------|-------|------------|------------|---------------|
| 0 | 10 | 40 | 4 | (up right, 100) | Atom | False | False | True | True | 0 |
| 1 | 10 | 40 | 4 | (down right, 100) | Atom | False | False | True | True | 0 |
| 2 | 10 | 40 | 4 | (up, 100) | Atom | False | False | True | True | 0 |
| 3 | 10 | 15 | 4 | (up right, 100) | Center | False | False | True | True | 0 |
| 4 | 10 | 15 | 4 | (down right, 100) | Center | False | False | True | True | 0 |
| 5 | 10 | 15 | 4 | (up, 100) | Center | False | False | True | True | 0 |
| 6–7 | 10 | 10 | 4 | (up right, 100) | Center | False | False | False | False | 0 |
| 8–9 | 10 | 10 | 4 | (down right, 100) | Center | False | False | False | False | 0 |
| 10–11 | 10 | 10 | 4 | (up, 100) | Center | False | False | False | False | 0 |
| 12–14 | 10 | 10 | 4 | (right, 100) | Center | False | False | False | False | 0 |
| 15–16 | 16 | 20 | 3 | (None, 100) | Atom | False | True | True | True | 0 |
| 17–18 | 25 | 15 | 3 | (None, 100) | Atom | False | True | True | True | 0 |
| 19–20 | 16 | 10 | 3 | (None, 100) | Center | False | True | True | True | 0 |
| 21–22 | 25 | 8 | 3 | (None, 100) | Center | False | True | True | True | 0 |
| 23–24 | 15 | 40 | 3 | (None, 0) | Atom | True | True | False | False | 0.85 |
| 25–26 | 15 | 40 | 3 | (None, 0) | Atom | True | True | False | False | 0.9 |
| 27–28 | 15 | 40 | 3 | (None, 0) | Atom | True | True | False | False | 0.95 |
| 29–30 | 25 | 40 | 3 | (None, 0) | Atom | True | True | False | False | 0.85 |
| 31–32 | 25 | 40 | 3 | (None, 0) | Atom | True | True | False | False | 0.9 |
| 33–34 | 25 | 40 | 3 | (None, 0) | Atom | True | True | False | False | 0.95 |
| 35–38 | 15 | 20 | 3 | (None, 0) | Center | True | True | False | False | 0.7 |
| 39–42 | 15 | 20 | 3 | (None, 0) | Center | True | True | False | False | 0.8 |
| 43–46 | 10 | 20 | 3 | (None, 0) | Center | True | True | False | False | 0.9 |
| 47–49 | 15 | 40 | 4 | (None, 0) | Atom | True | False | False | False | 0 |
| 50–52 | 25 | 40 | 4 | (None, 0) | Atom | True | False | False | False | 0 |
| 53–55 | 30 | 40 | 4 | (None, 0) | Atom | True | False | False | False | 0 |
| 56–58 | 50 | 40 | 4 | (None, 0) | Atom | True | False | False | False | 0 |

| | | | | | | | | | | |
|-------|----|----|---|------------------|--------|------|-------|-------|-------|---|
| 59–60 | 8 | 30 | 4 | (right, 1) | Atom | True | False | False | False | 0 |
| 61–62 | 8 | 30 | 4 | (up, 1) | Atom | True | False | False | False | 0 |
| 63–64 | 8 | 30 | 4 | (up right, 1) | Atom | True | False | False | False | 0 |
| 65–66 | 8 | 30 | 4 | (down right, 1) | Atom | True | False | False | False | 0 |
| 67–68 | 16 | 30 | 4 | (right, 1) | Atom | True | False | False | False | 0 |
| 69–70 | 16 | 30 | 4 | (up, 1) | Atom | True | False | False | False | 0 |
| 71–72 | 16 | 30 | 4 | (up right, 1) | Atom | True | False | False | False | 0 |
| 73–74 | 16 | 30 | 4 | (down right, 1) | Atom | True | False | False | False | 0 |
| 75–76 | 8 | 30 | 4 | (right, 1) | Center | True | False | False | False | 0 |
| 77–78 | 8 | 30 | 4 | (up, 1) | Center | True | False | False | False | 0 |
| 79–78 | 8 | 30 | 4 | (up right, 1) | Center | True | False | False | False | 0 |
| 81–82 | 8 | 30 | 4 | (down right, 1) | Center | True | False | False | False | 0 |
| 83–84 | 16 | 30 | 4 | (right, 1) | Center | True | False | False | False | 0 |
| 85–86 | 16 | 30 | 4 | (up, 1) | Center | True | False | False | False | 0 |
| 87–88 | 16 | 30 | 4 | (up right, 1) | Center | True | False | False | False | 0 |
| 89–90 | 16 | 30 | 4 | (down right, 1) | Center | True | False | False | False | 0 |
| 91 | 32 | 30 | 5 | (down, 1.2) | Center | True | False | False | False | 0 |
| 92 | 32 | 30 | 5 | (down left, 1.2) | Center | True | False | False | False | 0 |
| 93 | 32 | 30 | 5 | (left, 1.2) | Center | True | False | False | False | 0 |
| 94 | 32 | 30 | 4 | (down, 1.2) | Center | True | False | False | False | 0 |
| 95 | 32 | 30 | 4 | (down left, 1.2) | Center | True | False | False | False | 0 |
| 96 | 32 | 30 | 4 | (left, 1.2) | Center | True | False | False | False | 0 |
| 97 | 32 | 30 | 3 | (down, 1.2) | Center | True | False | False | False | 0 |
| 98 | 32 | 30 | 3 | (down left, 1.2) | Center | True | False | False | False | 0 |
| 99 | 32 | 30 | 3 | (left, 1.2) | Center | True | False | False | False | 0 |

Table B.1: Random walk parameters corresponding to the configurations shown in figure Fig. B.6 to B.9. All configurations have the default parameters: Periodic = True, Avoid clustering = 10.



Appendix C

Friction-strain profiles

Friction-strain profiles for the dataset for the Honeycomb (Fig. C.1), Tetrahedron (Fig. C.2) and Random walk (Fig. C.3) patterns respectively. The friction-strain profiles are computed as a mean over the three normal loads used in the dataset and plotted against the strain relative to the rupture strain. The legends show the names for each pattern and the corresponding (absolute) rupture strain. The data has been interpolated with a cubic spline interpolation in order to visualize the approximate shape of the continuous curve.

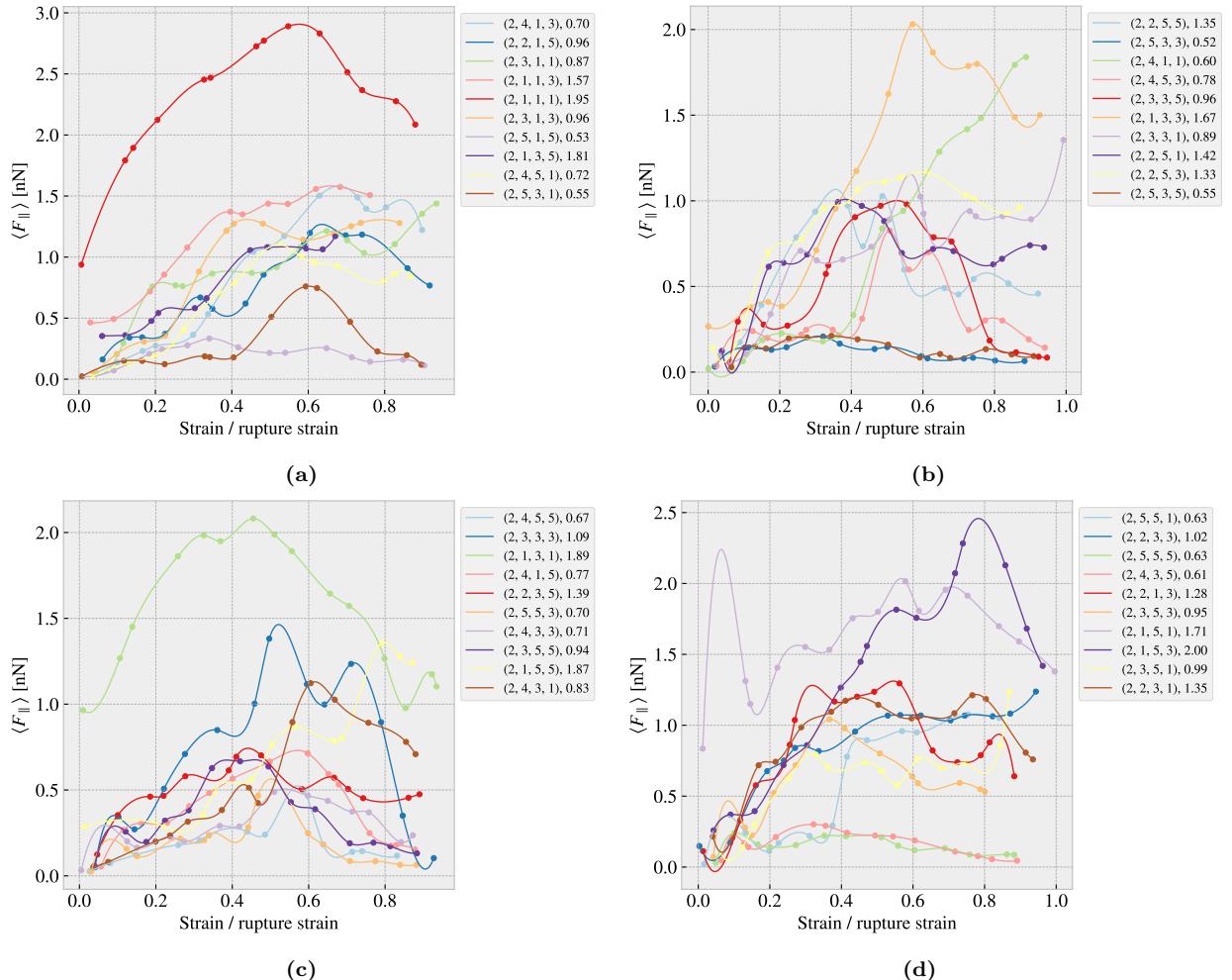


Figure C.1: Honeycomb friction-strain profiles.



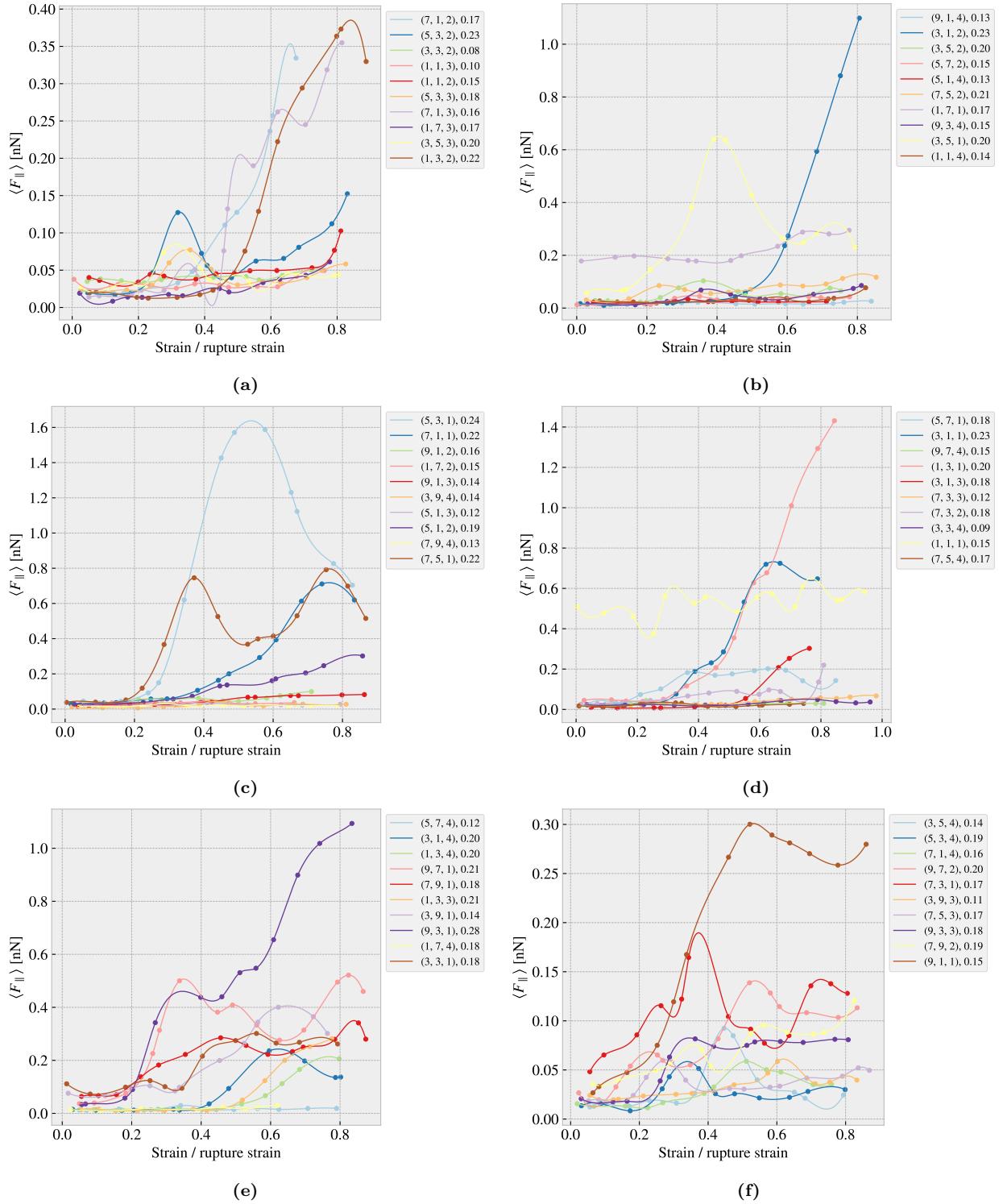


Figure C.2: Tetrahedron friction-strain profiles.

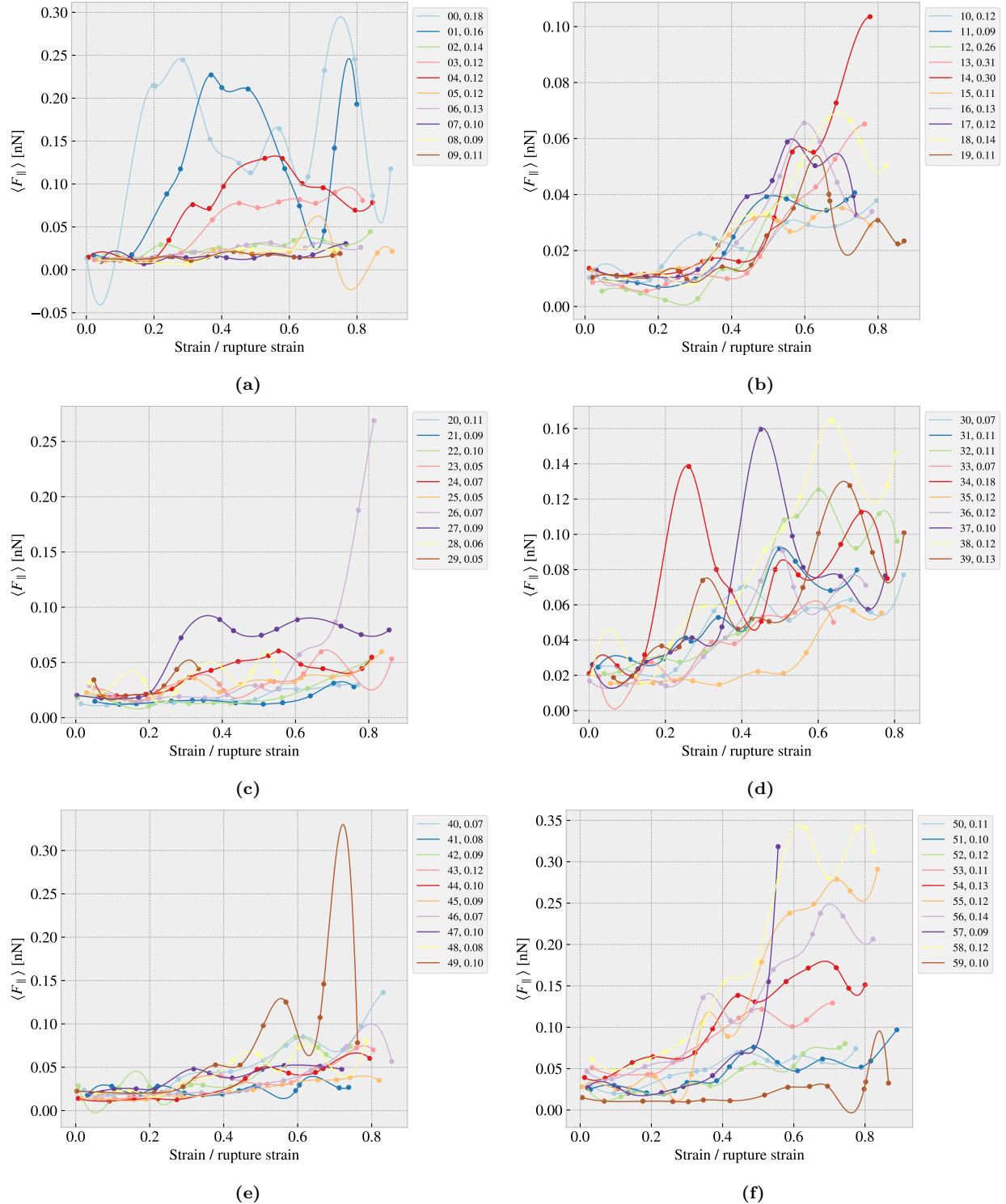


Figure C.3: Random walk friction-strain profiles. (The figure continues on the next page)



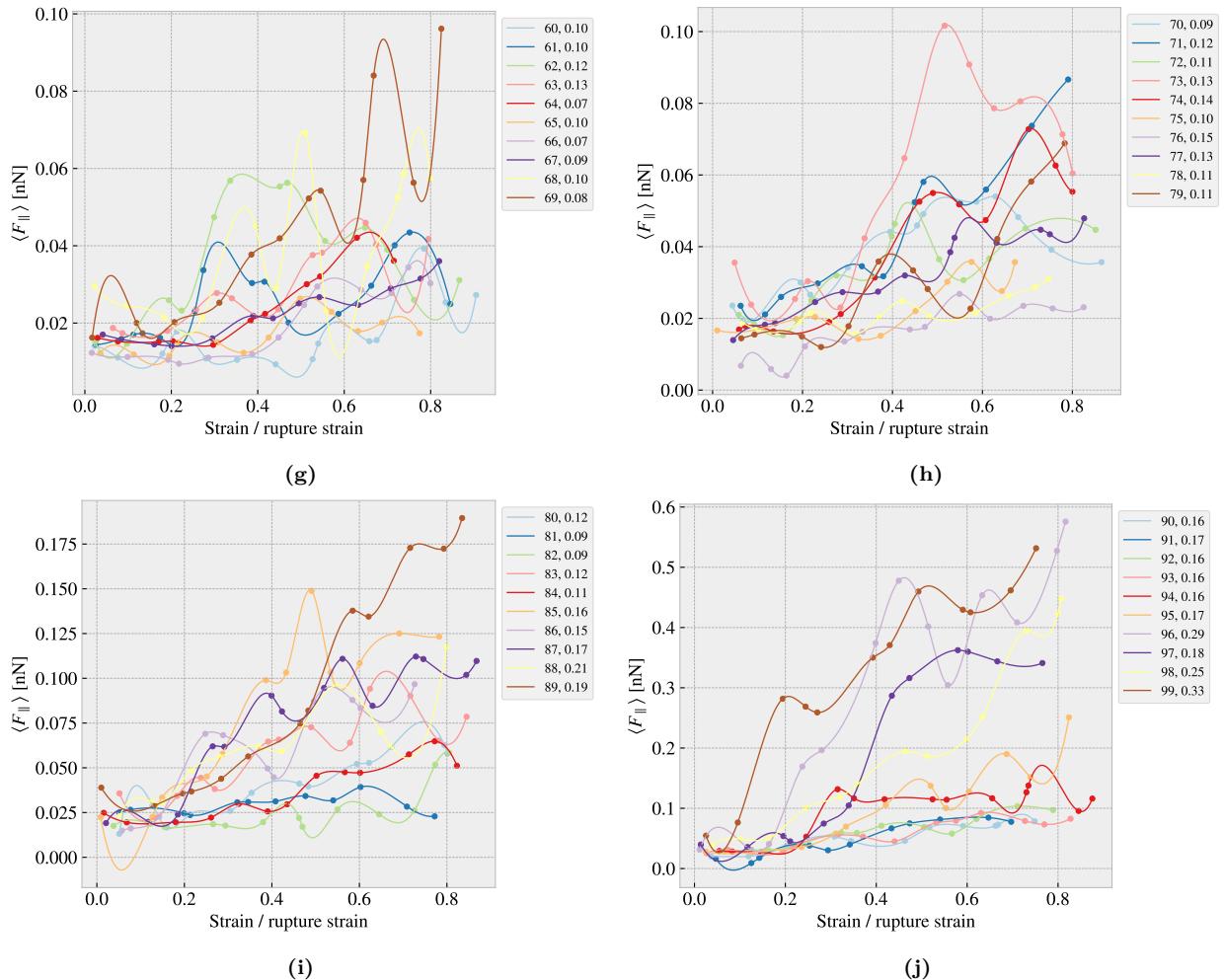


Figure C.3: Random walk friction-strain profiles.

Bibliography

- [1] E. Gnecco and E. Meyer, *Elements of friction theory and nanotribology* (Cambridge University Press, 2015).
- [2] Bhusnan, “Introduction”, in *Introduction to tribology* (John Wiley & Sons, Ltd, 2013) Chap. 1.
- [3] H.-J. Kim and D.-E. Kim, “Nano-scale friction: a review”, *International Journal of Precision Engineering and Manufacturing* **10**, 141–151 (2009).
- [4] K. Holmberg and A. Erdemir, “Influence of tribology on global energy consumption, costs and emissions”, *Friction* **5**, 263–284 (2017).
- [5] B. Bhushan, “Gecko feet: natural hairy attachment systems for smart adhesion – mechanism, modeling and development of bio-inspired materials”, in *Nanotribology and nanomechanics: an introduction* (Springer Berlin Heidelberg, Berlin, Heidelberg, 2008), pp. 1073–1134.
- [6] P. Z. Hanakata, E. D. Cubuk, D. K. Campbell, and H. S. Park, “Accelerated search and design of stretchable graphene kirigami using machine learning”, *Phys. Rev. Lett.* **121**, 255304 (2018).
- [7] P. Z. Hanakata, E. D. Cubuk, D. K. Campbell, and H. S. Park, “Forward and inverse design of kirigami via supervised autoencoder”, *Phys. Rev. Res.* **2**, 042006 (2020).
- [8] L.-K. Wan, Y.-X. Xue, J.-W. Jiang, and H. S. Park, “Machine learning accelerated search of the strongest graphene/h-bn interface with designed fracture properties”, *Journal of Applied Physics* **133**, 024302 (2023).
- [9] Y. Mao, Q. He, and X. Zhao, “Designing complex architectured materials with generative adversarial networks”, *Science Advances* **6**, eaaz4169 (2020).
- [10] Z. Yang, C.-H. Yu, and M. J. Buehler, “Deep learning model to predict complex stress and strain fields in hierarchical composites”, *Science Advances* **7**, eabd7416 (2021).
- [11] A. E. Forte, P. Z. Hanakata, L. Jin, E. Zari, A. Zareei, M. C. Fernandes, L. Sumner, J. Alvarez, and K. Bertoldi, “Inverse design of inflatable soft membranes through machine learning”, *Advanced Functional Materials* **32**, 2111610 (2022).
- [12] S. Chen, J. Chen, X. Zhang, Z.-Y. Li, and J. Li, “Kirigami/origami: unfolding the new regime of advanced 3D microfabrication/nanofabrication with “folding””, *Light: Science & Applications* **9**, 75 (2020).
- [13] OpenAI, *DALL-E2*, (2023) <https://openai.com/product/dall-e-2>.
- [14] Midjourney, *Midjourney*, (2023) <https://www.midjourney.com>.
- [15] Z. Deng, A. Smolyanitsky, Q. Li, X.-Q. Feng, and R. J. Cannara, “Adhesion-dependent negative friction coefficient on chemically modified graphite at the nanoscale”, *Nature Materials* **11**, 1032–1037 (2012).
- [16] B. Liu, J. Wang, S. Zhao, C. Qu, Y. Liu, L. Ma, Z. Zhang, K. Liu, Q. Zheng, and M. Ma, “Negative friction coefficient in microscale graphite/mica layered heterojunctions”, *Science Advances* **6**, eaaz6787 (2020).
- [17] D. Mandelli, W. Ouyang, O. Hod, and M. Urbakh, “Negative friction coefficients in superlubric graphite-hexagonal boron nitride heterojunctions”, *Phys. Rev. Lett.* **122**, 076102 (2019).
- [18] R. W. Liefferink, B. Weber, C. Coulais, and D. Bonn, “Geometric control of sliding friction”, *Extreme Mechanics Letters* **49**, 101475 (2021).
- [19] M. Metzsch, *Github repository*, <https://github.com/mikkkelme/MastersThesis>.

- [20] A. P. Thompson, H. M. Aktulga, R. Berger, D. S. Bolintineanu, W. M. Brown, P. S. Crozier, P. J. in 't Veld, A. Kohlmeyer, S. G. Moore, T. D. Nguyen, R. Shan, M. J. Stevens, J. Tranchida, C. Trott, and S. J. Plimpton, "LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales", *Comp. Phys. Comm.* **271**, 108171 (2022).
- [21] E. M. Nordhagen, *LAMMPS simulator*, <https://github.com/evenmn/lammps-simulator>.
- [22] A. H. Larsen, J. J. Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dułak, J. Friis, M. N. Groves, B. Hammer, C. Hargus, E. D. Hermes, P. C. Jennings, P. B. Jensen, J. Kermode, J. R. Kitchin, E. L. Kolsbjerg, J. Kubal, K. Kaasbjerg, S. Lysgaard, J. B. Maronsson, T. Maxson, T. Olsen, L. Pastewka, A. Peterson, C. Rostgaard, J. Schiøtz, O. Schütt, M. Strange, K. S. Thygesen, T. Vegge, L. Vilhelmsen, M. Walter, Z. Zeng, and K. W. Jacobsen, "The atomic simulation environment—a python library for working with atoms", *Journal of Physics: Condensed Matter* **29**, 273002 (2017).
- [23] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: an imperative style, high-performance deep learning library", in *Advances in neural information processing systems 32* (Curran Associates, Inc., 2019), pp. 8024–8035.
- [24] N Manini, O. M. Braun, E Tosatti, R Guerra, and A Vanossi, "Friction and nonlinear dynamics", *Journal of Physics: Condensed Matter* **28**, 293001 (2016).
- [25] B. Bhushan and A. V. Kulkarni, "Effect of normal load on microscale friction measurements", *Thin Solid Films* **278**, 49–56 (1996).
- [26] B. P. Hung, P. Y. Huri, J. P. Temple, A. Dorafshar, and W. L. Grayson, "Chapter 10 - craniofacial bone", in *3d bioprinting and nanotechnology in tissue engineering and regenerative medicine*, edited by L. G. Zhang, J. P. Fisher, and K. W. Leong (Academic Press, 2015), pp. 215–230.
- [27] J. Gao, W. D. Luedtke, D. Gourdon, M. Ruths, J. N. Israelachvili, and U. Landman, "Frictional forces and amontons' law: from the molecular to the macroscopic scale", *The Journal of Physical Chemistry B* **108**, Publisher: American Chemical Society, 3410–3425 (2004).
- [28] J. H. Dieterich, "Time-dependent friction in rocks", *Journal of Geophysical Research (1896-1977)* **77**, 3690–3697 (1972).
- [29] R. Burridge and L. Knopoff, "Model and theoretical seismicity", *Bulletin of the Seismological Society of America* **57**, 341–371 (1967).
- [30] O. M. Braun and J. Röder, "Transition from stick-slip to smooth sliding: an earthquakelike model", *Phys. Rev. Lett.* **88**, 096102 (2002).
- [31] W. I. Newman and A. M. Gabrielov, "Failure of hierarchical distributions of fibre bundles. I", *International Journal of Fracture* **50**, 1–14 (1991).
- [32] R. F. Smalley Jr., D. L. Turcotte, and S. A. Solla, "A renormalization group approach to the stick-slip behavior of faults", *Journal of Geophysical Research: Solid Earth* **90**, 1894–1900 (1985).
- [33] P. Selvadurai, P. Galvez, P. Mai, and S. Glaser, "Modeling frictional precursory phenomena using a wear-based rate- and state-dependent friction model in the laboratory", *Tectonophysics* **847**, 229689 (2023).
- [34] Y. Mo, K. T. Turner, and I. Szlufarska, "Friction laws at the nanoscale", *Nature* **457**, 1116–1119 (2009).
- [35] G. Carbone and F. Bottiglione, "Asperity contact theories: do they predict linearity between contact area and load?", *Journal of the Mechanics and Physics of Solids* **56**, 2555–2572 (2008).
- [36] F. Bowden and D. Tabor, *The friction and lubrication of solids*, International series of monographs on physics vb. 1 (Clarendon Press, 2001).
- [37] H.-J. Kim and D.-E. Kim, "Molecular dynamics simulation of atomic-scale frictional behavior of corrugated nano-structured surfaces", *Nanoscale* **4**, 3937–3944 (2012).
- [38] W. Commons, *File:asperities.svg — wikimedia commons, the free media repository*, (2023) <https://commons.wikimedia.org/w/index.php?title=File:Asperities.svg&oldid=659167170> (visited on 02/03/2023).
- [39] I. Szlufarska, M. Chandross, and R. W. Carpick, "Recent advances in single-asperity nanotribology", *Journal of Physics D: Applied Physics* **41**, 123001 (2008).

- [40] G. Binnig, C. F. Quate, and C. Gerber, “Atomic force microscope”, *Phys. Rev. Lett.* **56**, 930–933 (1986).
- [41] S. S. Perry, “Scanning Probe Microscopy Measurements of Friction”, *MRS Bulletin* **29**, 478–483 (2004).
- [42] Hertz, “On the contact of elastic solids”, *Crelle’s Journal* **92**, 156–171.
- [43] D. Maugis, “Adhesion of spheres: the jkr-dmt transition using a dugdale model”, *Journal of Colloid and Interface Science* **150**, 243–269 (1992).
- [44] M. H. Müser, “Rigorous field-theoretical approach to the contact mechanics of rough elastic solids”, *Phys. Rev. Lett.* **100**, 055504 (2008).
- [45] B. N. J. Persson, “Theory of rubber friction and contact mechanics”, *The Journal of Chemical Physics* **115**, 3840–3861 (2001).
- [46] J. A. Greenwood and J. B. P. Williamson, “Contact of nominally flat surfaces”, en, *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* **295**, 300–319 (1966).
- [47] A. Bush, R. Gibson, and T. Thomas, “The elastic contact of a rough surface”, *Wear* **35**, 87–111 (1975).
- [48] B. Luan and M. O. Robbins, “The breakdown of continuum models for mechanical contacts”, *Nature* **435**, 929–932 (2005).
- [49] Y. Dong, A. Vadakkepatt, and A. Martini, “Analytical models for atomic friction”, *Tribology Letters* **44**, 10.1007/s11249-011-9850-2 (2011).
- [50] E. Gnecco, R. Bennewitz, T. Gyalog, C. Loppacher, M. Bammerlin, E. Meyer, and H.-J. Güntherodt, “Velocity dependence of atomic friction”, *Phys. Rev. Lett.* **84**, 1172–1175 (2000).
- [51] P. Hänggi, P. Talkner, and M. Borkovec, “Reaction-rate theory: fifty years after kramers”, *Rev. Mod. Phys.* **62**, 251–341 (1990).
- [52] Y. Sang, M. Dubé, and M. Grant, “Thermal effects on atomic friction”, *Phys. Rev. Lett.* **87**, 174301 (2001).
- [53] S. Y. Krylov, K. B. Jinesh, H. Valk, M. Dienwiebel, and J. W. M. Frenken, “Thermally induced suppression of friction at the atomic scale”, *Phys. Rev. E* **71**, 065101 (2005).
- [54] S. Krylov and J. Frenken, “Thermal contact delocalization in atomic scale friction: a multitude of friction regimes”, English, *New Journal of Physics* **9**, 10.1088/1367-2630/9/10/398 (2007).
- [55] K. B. Jinesh, S. Y. Krylov, H. Valk, M. Dienwiebel, and J. W. M. Frenken, “Thermolubricity in atomic-scale friction”, *Phys. Rev. B* **78**, 155440 (2008).
- [56] M. H. Müser, “Nature of mechanical instabilities and their effect on kinetic friction”, *Phys. Rev. Lett.* **89**, 224301 (2002).
- [57] Q. Li, Y. Dong, D. Perez, A. Martini, and R. W. Carpick, “Speed dependence of atomic stick-slip friction in optimally matched experiments and molecular dynamics simulations”, *Phys. Rev. Lett.* **106**, 126101 (2011).
- [58] Y. Dong, Q. Li, and A. Martini, “Molecular dynamics simulation of atomic friction: a review and guide”, *Journal of Vacuum Science & Technology A* **31**, 030801 (2013).
- [59] K. Johnson and J. Woodhouse, “Stick-slip motion in the atomic force microscope”, *Tribology Letters* **5**, 155–160 (1998).
- [60] S. N. Medyanik, W. K. Liu, I.-H. Sung, and R. W. Carpick, “Predictions and observations of multiple slip modes in atomic-scale friction”, *Phys. Rev. Lett.* **97**, 136106 (2006).
- [61] A. Vanossi, N. Manini, M. Urbakh, S. Zapperi, and E. Tosatti, “Modeling friction: from nanoscale to mesoscale”, *Reviews of Modern Physics* **85**, 529–552 (2013).
- [62] J. Frenkel and T. Kontorova, “On the theory of plastic deformation and twinning”, *Phys. Z. Soviet.* **13** (1938).
- [63] J. Norell, A. Fasolino, and A. Wijn, “Emergent friction in two-dimensional frenkel-kontorova models”, *Physical Review E* **94**, 10.1103/PhysRevE.94.023001 (2016).
- [64] O. M. Braun and Y. S. Kivshar, “Nonlinear dynamics of the frenkel–kontorova model”, *Physics Reports* **306**, 1–108 (1998).
- [65] O. M. Braun, A. R. Bishop, and J. Röder, “Hysteresis in the underdamped driven frenkel-kontorova model”, *Phys. Rev. Lett.* **79**, 3692–3695 (1997).

- [66] M. Dienwiebel, N. Pradeep, G. S. Verhoeven, H. W. Zandbergen, and J. W. Frenken, “Model experiments of superlubricity of graphite”, *Surface Science* **576**, 197–211 (2005).
- [67] G. S. Verhoeven, M. Dienwiebel, and J. W. M. Frenken, “Model calculations of superlubricity of graphite”, *Phys. Rev. B* **70**, 165418 (2004).
- [68] C. Kittel, *Introduction to solid state physics*, 8th ed. (Wiley, 2004).
- [69] J. A. van den Ende, A. S. de Wijn, and A. Fasolino, “The effect of temperature and velocity on superlubricity”, *Journal of Physics: Condensed Matter* **24**, 445009 (2012).
- [70] X.-H. Zhang, U. Tartaglino, G. E. Santoro, and E. Tosatti, “Velocity plateaus and jumps in carbon nanotube sliding”, *Surface Science* **601**, 3693–3696 (2007).
- [71] X. H. Zhang, G. E. Santoro, U. Tartaglino, and E. Tosatti, “Dynamical chiral symmetry breaking in sliding nanotubes”, *Phys. Rev. Lett.* **102**, 125502 (2009).
- [72] M. Weiss and F.-J. Elmer, “Dry friction in the Frenkel-Kontorova-Tomlinson model: dynamical properties”, *Zeitschrift für Physik B Condensed Matter* **104**, 55–69 (1997).
- [73] B. Bhushan, “Nanotribology and nanomechanics”, *Wear* **259**, 15th International Conference on Wear of Materials, <https://doi.org/10.1016/j.wear.2005.01.010> (2005).
- [74] O. Penkov, H.-J. Kim, H.-J. Kim, and D.-E. Kim, “Tribology of graphene: A review”, *International Journal of Precision Engineering and Manufacturing* **15**, 577–585 (2014).
- [75] X. Zhao, M. Hamilton, W. G. Sawyer, and S. S. Perry, “Thermally activated friction”, *Tribology Letters* **27**, 113–117 (2007).
- [76] G. Paolicelli, M. Tripathi, V. Corradini, A. Candini, and S. Valeri, “Nanoscale frictional behavior of graphene on sio₂ and ni(111) substrates”, *Nanotechnology* **26**, 055703 (2015).
- [77] S. Zhang, Y. Hou, S. Li, L. Liu, Z. Zhang, X.-Q. Feng, and Q. Li, “Tuning friction to a superlubric state via in-plane straining”, *Proceedings of the National Academy of Sciences* **116**, Publisher: Proceedings of the National Academy of Sciences, 24452–24456 (2019).
- [78] H. M. Yoon, Y. Jung, S. C. Jun, S. Kondaraju, and J. S. Lee, “Molecular dynamics simulations of nanoscale and sub-nanoscale friction behavior between graphene and a silicon tip: analysis of tip apex motion.”, *Nanoscale* **7** 14, 6295–303 (2015).
- [79] S. Li, Q. Li, R. W. Carpick, P. Gumbsch, X. Z. Liu, X. Ding, J. Sun, and J. Li, “The evolving quality of frictional contact with graphene”, *Nature* **539**, Number: 7630, 541–545 (2016).
- [80] A. Wijn, A. Fasolino, A. Filippov, and M. Urbakh, “Low friction and rotational dynamics of crystalline flakes in solid lubrication”, *Europhysics Letters (epl)* **95**, 10.1209/0295-5075/95/66002 (2011).
- [81] X. Feng, S. Kwon, J. Y. Park, and M. Salmeron, “Superlubric sliding of graphene nanoflakes on graphene”, *ACS Nano* **7**, Publisher: American Chemical Society, 1718–1724 (2013).
- [82] F. Bonelli, N. Manini, E. Cadelano, and L. Colombo, “Atomistic simulations of the sliding friction of graphene flakes”, *The European Physical Journal B* **70**, 449–459 (2009).
- [83] M. Reguzzoni, A. Fasolino, E. Molinari, and M. C. Righi, “Friction by shear deformations in multilayer graphene”, *The Journal of Physical Chemistry C* **116**, 21104–21108 (2012).
- [84] Y. Liu, F. Grey, and Q. Zheng, “The high-speed sliding friction of graphene and novel routes to persistent superlubricity”, *Scientific Reports* **4**, 4875 (2014).
- [85] P. Zhu and R. Li, “Study of nanoscale friction behaviors of graphene on gold substrates using molecular dynamics”, *Nanoscale Research Letters* **13**, 34 (2018).
- [86] J. Zhang, E. Osloub, F. Siddiqui, W. Zhang, T. Ragab, and C. Basaran, “Anisotropy of graphene nanoflake diamond interface frictional properties”, *Materials* **12**, 10.3390/ma12091425 (2019).
- [87] L. Colombo, “Tight-binding molecular dynamics: A primer”, *La Rivista del Nuovo Cimento* **28**, 1–59 (2005).
- [88] A. Sircar and P. K. Patra, “A simple generalization of Prandtl–Tomlinson model to study nanoscale rolling friction”, *Journal of Applied Physics* **127**, 135102, 10.1063/1.5143062 (2020).
- [89] M. Reguzzoni and M. C. Righi, “Size dependence of static friction between solid clusters and substrates”, *Phys. Rev. B* **85**, 201412 (2012).
- [90] N. Varini, A. Vanossi, R. Guerra, D. Mandelli, R. Capozza, and E. Tosatti, “Static friction scaling of physisorbed islands: the key is in the edge”, *Nanoscale* **7**, 2093–2101 (2015).

- [91] T. Filleter, J. L. McChesney, A. Bostwick, E. Rotenberg, K. V. Emtsev, T. Seyller, K. Horn, and R. Bennewitz, “Friction and dissipation in epitaxial graphene films”, *Phys. Rev. Lett.* **102**, 086102 (2009).
- [92] C. Lee, Q. Li, W. Kalb, X. Liu, H. Berger, R. Carpick, and J. Hone, “Frictional characteristics of atomically thin sheets”, *Science (New York, N.Y.)* **328**, 76–80 (2010).
- [93] E. Thormann, “Negative friction coefficients”, *Nature Materials* **12**, 468–468 (2013).
- [94] O Zwörner, H Hölscher, U. Schwarz, and R Wiesendanger, “The velocity dependence of frictional forces in point-contact friction”, *APPLIED PHYSICS A MATERIALS SCIENCE AND PROCESSING* **66**, S263–S268 (1998).
- [95] R. Guerra, U. Tartaglino, A. Vanossi, and E. Tosatti, “Ballistic nanofriction”, *Nature Materials* **9**, 634–637 (2010).
- [96] P. Carloni, U. Rothlisberger, and M. Parrinello, “The Role and Perspective of Ab Initio Molecular Dynamics in the Study of Biological Systems”, *Accounts of Chemical Research* **35**, Publisher: American Chemical Society, 455–464 (2002).
- [97] P. Hohenberg and W. Kohn, “Inhomogeneous electron gas”, *Phys. Rev.* **136**, B864–B871 (1964).
- [98] J. Tersoff, “New empirical approach for the structure and energy of covalent systems”, *Phys. Rev. B* **37**, 6991–7000 (1988).
- [99] F. H. Stillinger and T. A. Weber, “Computer simulation of local order in condensed phases of silicon”, *Phys. Rev. B* **31**, 5262–5271 (1985).
- [100] Q. Zhang, D. Diao, and M. Kubo, “Nanoscratching of multi-layer graphene by molecular dynamics simulations”, *Tribology International* **88**, 85–88 (2015).
- [101] S. Corporation, *Pair_style lj/cut command*, (Dec. 22, 2022) https://docs.lammps.org/pair_lj.html (visited on 01/24/2023).
- [102] X. Wang, S. Ramírez-Hinestrosa, J. Dobnikar, and D. Frenkel, “The lennard-jones potential: when (not) to use it”, *Phys. Chem. Chem. Phys.* **22**, 10624–10633 (2020).
- [103] R. Naeem, *Lennard-jones potential*, (Nov. 25, 2022) [https://chem.libretexts.org/Bookshelves/Physical_and_Theoretical_Chemistry_Textbook_Maps/Supplemental_Modules_\(Physical_and_Theoretical_Chemistry\)/Physical_Properties_of_Matter/Atomic_and_Molecular_Properties/Intermolecular_Forces/Specific_Interactions/Lennard-Jones_Potential](https://chem.libretexts.org/Bookshelves/Physical_and_Theoretical_Chemistry_Textbook_Maps/Supplemental_Modules_(Physical_and_Theoretical_Chemistry)/Physical_Properties_of_Matter/Atomic_and_Molecular_Properties/Intermolecular_Forces/Specific_Interactions/Lennard-Jones_Potential) (visited on 01/24/2023).
- [104] S. Corporation, *Pair_style sw command*, (Dec. 22, 2022) https://docs.lammps.org/pair_sw.html (visited on 01/24/2023).
- [105] S. Corporation, *Pair_style tersoff command*, (Dec. 22, 2022) https://docs.lammps.org/pair_tersoff.html (visited on 01/24/2023).
- [106] W. Commons, *File:bond-order interatomic potential.png — wikimedia commons, the free media repository*, (2023) https://commons.wikimedia.org/wiki/File:Bond-order_interatomic_potential.png (visited on 05/02/2023).
- [107] J. Tersoff, “Modeling solid-state chemistry: interatomic potentials for multicomponent systems”, *Phys. Rev. B* **39**, 5566–5568 (1989).
- [108] D. Frenkel and B. Smit, *Understanding molecular simulation: from algorithms to applications*, Second, Vol. 1, Computational Science Series (Academic Press, San Diego, 2002).
- [109] F. Ravndal and E. G. Flekkøy, *Statistical physics— a second course*, (Mar. 2019) https://www.uio.no/studier/emner/matnat/fys/FYS4130/v19/pensumliste/stat-phys_2019.pdf.
- [110] T. Schneider and E. Stoll, “Molecular-dynamics study of a three-dimensional one-component model for distortive phase transitions”, *Phys. Rev. B* **17**, 1302–1322 (1978).
- [111] J. Lederer, “Activation functions in artificial neural networks: a systematic overview”, (2021).
- [112] P. Shankar, “A review on artificial neural networks”, **3**, 166–169 (2022).
- [113] A. Binder, *Lecture materials: IN5400 — Machine Learning for Image Analysis*, (2022) https://www.uio.no/studier/emner/matnat/ifi/IN5400/v22/lecture-materials/in5400_2022_slides_lecture2.pdf (visited on 05/08/2023).
- [114] N. C. Thompson, K. Greenewald, K. Lee, and G. F. Manso, “The computational limits of deep learning”, (2022).

- [115] J. G. Sam Lau and D. Nolan, *Principles and techniques of data science, 11.4. stochastic gradient descent*, (2020) https://www.samlau.me/test-textbook/ch/11/gradient_stochastic.html (visited on 05/08/2023).
- [116] D. P. Kingma and J. Ba, “Adam: a method for stochastic optimization”, (2017).
- [117] T. Salimans and D. P. Kingma, “Weight normalization: A simple reparameterization to accelerate training of deep neural networks”, *CoRR abs/1602.07868* (2016).
- [118] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: surpassing human-level performance on imagenet classification”, *IEEE International Conference on Computer Vision (ICCV 2015)* **1502**, [10.1109/ICCV.2015.123](https://doi.org/10.1109/ICCV.2015.123) (2015).
- [119] S. Ioffe and C. Szegedy, “Batch normalization: accelerating deep network training by reducing internal covariate shift”, in Proceedings of the 32nd international conference on international conference on machine learning - volume 37, ICML’15 (2015), 448–456.
- [120] L. N. Smith, “A disciplined approach to neural network hyper-parameters: part 1 - learning rate, batch size, momentum, and weight decay”, *ArXiv abs/1803.09820* (2018).
- [121] B. Z. Cunha, C. Droz, A. Zine, S. Foulard, and M. Ichchou, “A Review of Machine Learning Methods Applied to Structural Dynamics and Vibroacoustic”, working paper or preprint, Apr. 2022.
- [122] S. Saha, *A comprehensive guide to convolutional neural networks — the eli5 way*, (2018) <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53> (visited on 05/08/2023).
- [123] L. Guerdan, *Diving into temporal convolutional networks*, (2019) <https://lukeguerdan.com/blog/2019/intro-to-tcns/> (visited on 05/08/2023).
- [124] G. Cybenko, “Approximation by superpositions of a sigmoidal function”, *Mathematics of Control, Signals and Systems* **2**, 303–314 (1989).
- [125] Y. Bengio, “Practical recommendations for gradient-based training of deep architectures”, in *Neural networks: tricks of the trade: second edition*, edited by G. Montavon, G. B. Orr, and K.-R. Müller (Springer Berlin Heidelberg, Berlin, Heidelberg, 2012), pp. 437–478.
- [126] R. Holbrook and A. Cook, *Overfitting and underfitting*, <https://www.kaggle.com/code/ryanholbrook/overfitting-and-underfitting> (visited on 05/08/2023).
- [127] JavaTpoint., *Gradient descent in machine learning*, <https://www.javatpoint.com/gradient-descent-in-machine-learning> (visited on 05/08/2023).
- [128] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: visual explanations from deep networks via gradient-based localization”, *International Journal of Computer Vision* **128**, 336–359 (2019).
- [129] S. Katoch, S. S. Chauhan, and V. Kumar, “A review on genetic algorithm: past, present, and future”, *Multimedia Tools and Applications* **80**, 8091–8126 (2021).
- [130] R Jiang, K. Szeto, Y. Luo, and D. Hu, “Distributed parallel genetic algorithm with path splitting scheme for the large traveling salesman problems”, in Proceedings of conference on intelligent information processing, 16th world computer congress (2000), pp. 21–25.
- [131] K. Szeto, K. Cheung, and S. Li, “Effects of dimensionality on parallel genetic algorithms”, in Proceedings of the 4th international conference on information system, analysis and synthesis, orlando, florida, usa, Vol. 2 (1998), pp. 322–325.
- [132] K. Y. Szeto and L. Fong, “How adaptive agents in stock market perform in the presence of random news: a genetic algorithm approach”, in Intelligent data engineering and automated learning—ideal 2000. data mining, financial engineering, and intelligent agents: second international conference shatin, nt, hong kong, china, december 13–15, 2000 proceedings 2 (Springer, 2000), pp. 505–510.
- [133] K. L. Shiu and K. Y. Szeto, “Self-adaptive mutation only genetic algorithm: an application on the optimization of airport capacity utilization”, in Intelligent data engineering and automated learning—ideal 2008: 9th international conference daejeon, south korea, november 2-5, 2008 proceedings 9 (Springer, 2008), pp. 428–435.
- [134] G. Wang, C. Chen, and K. Y. Szeto, “Accelerated genetic algorithms with markov chains”, in *Nature inspired cooperative strategies for optimization (nicso 2010)*, edited by J. R. González, D. A. Pelta, C. Cruz, G. Terrazas, and N. Krasnogor (Springer Berlin Heidelberg, Berlin, Heidelberg, 2010), pp. 245–254.

- [135] D. Gray, A. McCaughan, and B. Mookerji, “Crystal structure of graphite, graphene and silicon”, Physics for Solid State Applications **6**, 730 (2009).
- [136] L. Burrows, *New pop-up strategy inspired by cuts, not folds*, (Feb. 24, 2017) <https://seas.harvard.edu/news/2017/02/new-pop-strategy-inspired-cuts-not-folds>.
- [137] *Scotch cushion lock protective wrap*, https://www.scotchbrand.com/3M/en_US/scotch-brand/products/catalog/~/Scotch-Cushion-Lock-Protective-Wrap/?N=4335+3288092498+3294529207&rt=rud.
- [138] H. Tomaç, Z. Guchan, and N. Altun, “How the stiletto heeled shoes which are popularly preferred by many women affect balance and functional skills?”, Health Care for Women International **43**, 1–11 (2020).
- [139] K. Gibbs, *Pressure*, (2020) https://www.schoolphysics.co.uk/age16-19/Mechanics/Statics/text/Pressure_/index.html.
- [140] F. B. Ltd, *Foot facts*, (2023) <https://www.footbionics.com/Patients/Foot+Facts.html>.
- [141] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition”, CoRR **abs/1409.1556** (2014).
- [142] Neurohive, *VGG16 -- Convolutional Network for Classification and Detection*, (2018) <https://neurohive.io/en/popular-networks/vgg16/> (visited on 05/07/2023).