

Tuning Frictional Properties of Kirigami Altered Graphene Sheets using Molecular Dynamics and Machine Learning

Designing a Negative Friction Coefficient

Mikkel Metzsch Jensen



Thesis submitted for the degree of
Master in Computational Science: Materials Science
60 credits

Department of Physics
Faculty of mathematics and natural sciences

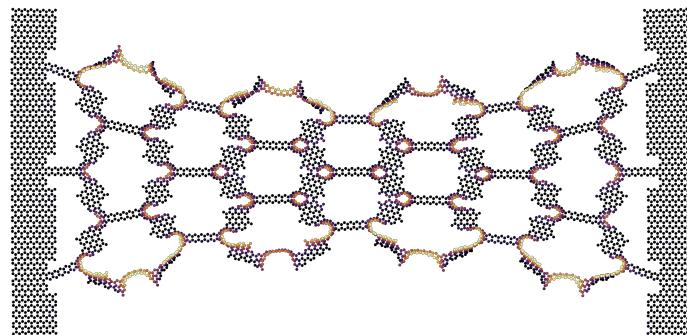
UNIVERSITY OF OSLO

Spring 2023

Tuning Frictional Properties of Kirigami Altered Graphene Sheets using Molecular Dynamics and Machine Learning

Designing a Negative Friction Coefficient

Mikkel Metzsch Jensen



© 2023 Mikkel Metzsch Jensen

Tuning Frictional Properties of Kirigami Altered Graphene Sheets using Molecular Dynamics and Machine Learning

<http://www.duo.uio.no/>

Printed: Reprocentralen, University of Oslo

Abstract

Abstract.

Acknowledgments

Acknowledgments.

List of Symbols

F_N Normal force (normal load)

Acronyms

CNN convolutional neural network. 17, 18

FC fully connected. 18

MD molecular dynamics. 2, 3

ML machine learning. 2, 3

MSE mean squared error. 18

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Approach	3
1.3	Contributions	3
1.4	Thesis structure	3
I	Background Theory	5
2	Machine Learning	7
2.1	Neural network	7
2.2	Convolutional neural network	7
2.3	Optimizers	7
2.4	Hypertuning strategies	7
2.5	Prediction explanation	7
II	Simulations	9
3	Dataset study	11
3.1	Generating data	11
3.2	Data analysis	12
3.3	Properties of interest	13
3.4	Machine learning	17
3.4.1	Architecture	17
3.4.2	Data handling	18
3.4.2.1	Input	18
3.4.2.2	Output	18
3.4.2.3	Data augmentation	18
3.4.3	Loss	18
3.4.4	Hypertuning	19
3.5	Accelerated Search	22
3.5.1	Markov-Chain Accelerated Genetic Algorithms	22
3.5.1.1	Talk about traditional method also?	22
3.5.1.2	Implementing for 1D chromosome (following article closely)	22
Appendices		25
Appendix A		27
Appendix B		29
Appendix C		31

Chapter 1

Introduction

Structure of Motivation section:

1. Introduce and motivate friction broadly.
2. Motives for friction control using a grasping robot as example.
3. Analog to gecko feet where adhesive properties are turned on and off.
4. Interest in origin of friction through nanoscale studies which further motivates the use of MD.
5. Intro to metamaterials and the use of kirigami designs,
6. How to optimize kirigami designs with reference to Hanakata and motivating the use of ML.
7. Out-of-plane buckling motivates the use of kirigami for frictional properties.

Does some of the latter paragraphs belong to the approach section?

1.1 Motivation

Friction is a fundamental force that takes part in most of all interactions with physical matter. Even though the everyday person might not be familiar with the term *friction* we recognize it as the inherent resistance to sliding motion. Some surfaces appear slippery and some 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 [p. 5] [1]. It is probably safe to say that the concept of friction is integrated in our everyday life to such an extent that most people take it for granted. However, the efforts to control friction dates 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” and 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, reducing friction is of great interest as this has tremendous advantages for energy efficiency. It has been reported that tribological problems have a significant potential for 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 a finetuned object handling is required. While achieving a certain “constant” friction response is readily obtained through appropriate material choices during manufacturing, we are yet to unlock the capabilities to alter friction dynamically on the go. One example from nature inspiring us to think along these lines are 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 geckos are able to produce large adhesive forces, they retain the ability to remove their 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 it 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.

In the recent years an increasing amount of interest has gone into the studies of the microscopic origin 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 dictates machining characteristics [3]. With concurrent progress in computational power and development of Molecular Dynamics (MD), numerical investigations serve as an extremely useful 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 material compositions are designed meticulously to enhance certain physical properties [6][7][8][9][10][11]. This is often achieved either by intertwining different material types or removing certain regions completely. In recent papers by Hanakata et al. [6](2018) [7](2020) numerical studies have showcased that mechanical properties of a graphene sheet, in this case 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 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 intractable problems ruling out analytic solutions.

Earlier architecture 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 simulation time on modern day hardware. However, this computational time constraint can be relaxed by the use of machine learning (ML) which have 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 all together resulting in an *accelerated search* of designs. This can be further improved by guiding the search accordingly to the most promising candidates, as for instance done with the *genetic algorithm* which suggest new designs based on mutation and crossing of the best candidates so far. Another, even more sophisticated approach, is through generative methods such as *Generative Adversarial Networks* (GAN). By working with a so-called *encoder-decoder* network structure, one can build a model that reverses the prediction process. That is, the model predicts a design from a set of physical target properties. In the papers by Hanakata et al. 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. attributes the variety in yield 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 the kirigami cut and stretch procedure can also be exploited for the design of frictional metamaterials. For certain designs we might hope to find a relationship between stretching of the sheet and frictional properties. If significant, this could give rise to a variability of the friction response beyond manufacturing material choice. 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; Slippery and smooth when in contact with people and rough and firmly gripping when moving heavy objects. In addition, a possible coupling between stretch and the normal load through a nanomachine design would allow for an altered friction coefficient. This invites the idea of non-linear friction coefficients which might in theory also take on negative values given the right response from stretching. The latter would constitute an extremely rare property. This has (**only?**) been reported indirectly for bulk graphite by Deng et al. [13] where the friction kept increasing during the unloading phase. **Check for other cases and what I can really say here.**

To the best of our knowledge, kirigami has not yet been implemented to alter the frictional properties of a nanoscale system. In a recent paper by Liefferink et al. [14](2021) it is reported that macroscale kirigami can be used to dynamically control the macroscale roughness of a surface through stretching which was used to change the frictional coefficient by more than one order of magnitude. This supports the idea that kirigami designs can in fact be used to alter friction, but we believe that taking this concept to the nanoscale regime would involve a different set of underlying mechanisms and thus contribute to new insight in this field.

1.2 Approach

In this thesis we investigate the possibility to alter and control the frictional properties of a graphene sheet through application of kirigami inspired cuts and stretching of the sheet. With the use of MD simulations we evaluate the friction properties under different physical conditions in order to get insight into the prospects of this field. By evaluating variations of two kirigami inspired patterns and a series of random walk generated patterns we create a dataset containing information of the frictional properties associated with each design under different load and stretch conditions. We apply ML to the dataset and use an accelerated search approach to optimize for different properties of interest. The subtask of the thesis are presented more comprehensively in the following.

1. Define a sheet indexing that allows for a unique mapping of patterns between a hexagonal graphene lattice representation to a matrix representation suited for numerical analysis.
2. Design a MD simulation procedure to evaluate the frictional properties of a given graphene sheet under specified physical conditions such as load, stretch, temperature etc.
3. Find and implement suitable kirigami patterns which exhibit out-of-plane buckling under tensile load. This includes the creation of a framework for creating variations within each pattern class. Additionally create a procedure for generating different styles of random walk patterns.
4. Perform a pilot study of a representative subset of patterns in order to determine appropriate simulation parameters to use for the further study along with an analysis of the frictional properties shown in the subset.
5. Create a dataset consisting of the chosen kirigami variations and random walk patterns and analyse data trends.
6. Train a neural network to map from the design space to physical properties such as mean friction, maximum friction, contact area etc. and evaluate the performance.
7. Perform an accelerated search optimizing for interesting frictional properties using the ML model. This should be done both through the pattern generation procedures and by following a genetic algorithm approach.
8. Use the most promising candidates from the accelerated search to investigate the prospects of creating a nanomachine setup which exhibits a negative friction coefficient.
9. Study certain designs of interest with the scope of revealing underlying mechanism. This includes simple correlation analysis but also a visualization of feature and gradient maps of the ML network.

Is the list of subtask too specific? Some of the details here might be better suited for the thesis structure section.

1.3 Contributions

What did I actually achieve

1.4 Thesis structure

How is the thesis structured.

Part I

Background Theory

Chapter 2

Machine Learning

2.1 Neural network

Feed forward fully connected network.

2.2 Convolutional neural network

2.3 Optimizers

2.4 Hypertuning strategies

2.5 Prediction explanation

Looking at feature maps and gradient maps.

Part II

Simulations

Chapter 3

Dataset study

3.1 Generating data

The dataset consist of friction simulations of various cut configurations and combinations of normal load and stretch. For each configuration we sample 15 pseudo uniform (refer to relevant section here) strecth values between zero and the rupture stretch found in the rupture test. The normal force is uniformly sampled in the range $[0.1, 10]$ nN. In total this gives 3×15 data points for each configuration. For the remaining parameters we use the values presented in the pilot study (see ??). We generate 68 configurations of the Tetrahedron pattern type, 45 of the Honeycomb type and 100 of the Random walk type. A summary of the dataset is given in Table 3.1 while all configurations are shown explicitly in ???. Notice that not all submitted data points “makes it” to the final dataset. This is due a small variation in rupture stretch points which were not anticipated during the creation of the numerical framework for submitting multiple simulation. After performing the rupture test the simulation is restarted with a new substrate size corresponding to the measured rupture stretch limit and also with new random velocity and thermostat initializations values. The sheet is then stretched and checkpoints of the simulation state (LAMMPS restart files) are saved for each of the targeted stretch samples. However, if the rupture points arrives slightly early than syggested by the rupture test, some sampled stretch values might not get a corresponding checkpoint file. Thus, these data points are not included in the data set even though they ideally should have been noted as a rupture event. This could quite easily have been mittigated by a rewrite of that part of the code, but it was first discovered after the dataset had been created. However, the dataset still includes 11.57 % rupture events and it most likely that the most cases with a lost rupturer event have a rupture event stored for the preeceding stretch value instead which captures the information of the sheet stretch limit on its own.

Table 3.1: Summary of the number of generated data points in the dataset. Due to slight deviations in the rupture stretch and the specific numerical procedure not all submitted simulations “makes it” to the final dataset. Notice that the Tetrahedon (7, 5, 2) and Honeycomb (2, 2, 1, 5) from the pilot study is rerun as a part of the Tetrahedon and the Honeycomb datasets seperately. In the latter datasets the reference point for the pattern is randomized and thus theese configurations is not fully identical. This is the idea behind the difference of 2 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 %)

3.2 Data analysis

In order to gain insight into the correlations between variables associated to the simulations we calculate the correlations coefficients between all variable combinations. More specific, we are going to calculate the Pearson product-moment correlation coefficient (PPMCC) for which is defined, between data set X and Y , as

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

where $\text{Cov}(X, Y)$ is the covariance, μ the mean value and σ the standard deviation. The correlation coefficients ranges from perfect negative correlation (-1) through no correlation (0) to a perfect positive correlation (1). The correlation coefficients is shown in Fig. 3.1

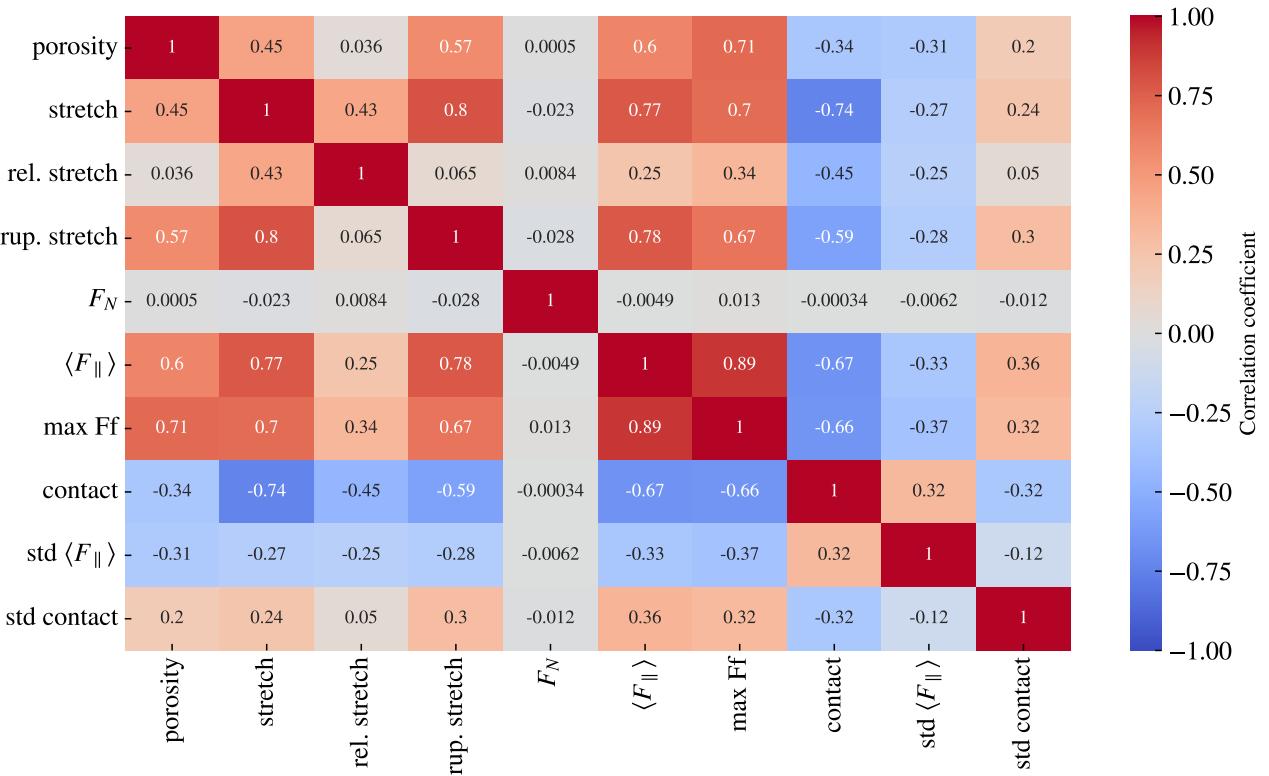


Figure 3.1: Pearson product-moment correlation coefficients for the full dataset (see Table 3.1).

From Fig. 3.1 we especially notice that the mean friction force $\langle F_{||} \rangle$ has a significant positively correlation with stretch (0.77) and porosity (0.60) (void fraction). However, the relative stretch, which is scaled by the rupture stretch, has a weaker correlation of only 0.25 which indicates that it is the absolute stretch value that has the most significant impact on the friction force increase during stretching. This is further supported by the fact that the mean friction and the rupture stretch is also strongly positively correlated (0.78). From figure Fig. 3.1 we also observe that the contact bond count is negatively correlated with the mean friction (-0.67) and the stretch value (-0.74) which is consistent with the trend observed in the pilot study ?? and ?? of the contact decreasing with increasing stretch and mean friction. However, we must take note that the correlation coefficients is a measure of the strength and slope of a forced linear fit on the data. We clearly observed a non-linear relationship between stretch and mean friction for the tetrahedron and honeycomb pattern used in the pilot study ?? where the relationship was partwise characterized by a positive correlation for some stretch ranges and partwise negative correlation for other stretch ranges. Hence, interesting strong regime-specific correlations might not be accurately highlighted by the correlation coefficients shown in Fig. 3.1.

In Fig. 3.2 we have visualized the data (excluding the pilot study) for chosen pairs of variables on the axes. In addition to a visual confirmation of how the given correlations look in a 2D plot we also get a feeling for the

coverage in various areas of the parameter space that we are eventually going to feed the neural network. The honeycomb pattern is spanning a significant larger range of stretch, contact and mean friction makes the data rather biased towards the Honeycomb pattern in those areas.

Uncommented to decrease loading time

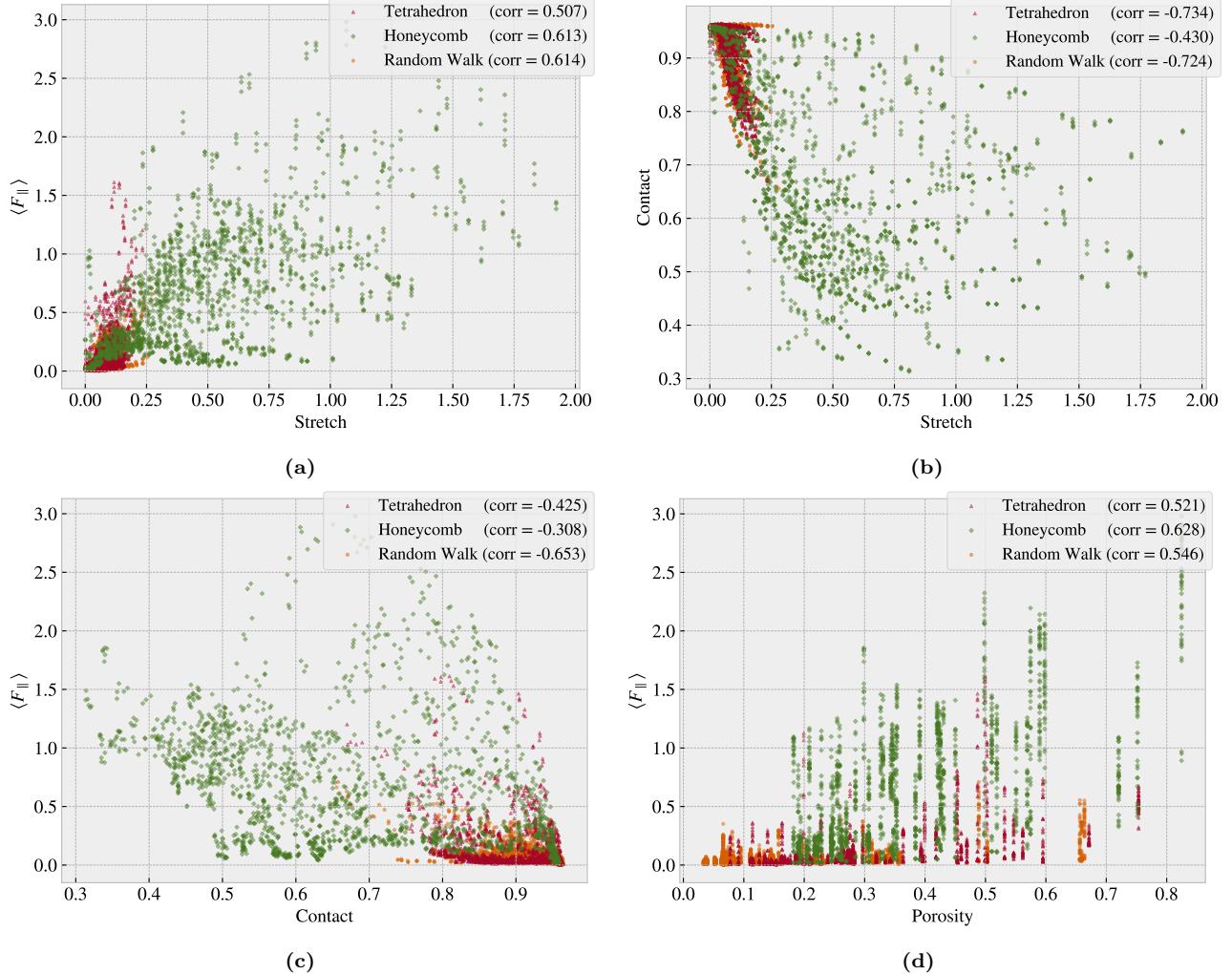


Figure 3.2: Scatter plot of the data sets Tetrahedron, Honeycomb and Random Walk (excluding the pilot study) for various variable combinations in order to visualize some chosen correlations of interest and distributions in the data

3.3 Properties of interest

From the Pilot study we discovered that it might be possible to achieve a negative friction coefficient for certain kirigami cut configurations under the assumption of a system with coupled normal force F_N and stretch S . This stands as the main property of interest to explore further in the dataset. However, it is not obvious how one should quantify this in a rigorous manner. The friction coefficient is by our definition (see theory sec XXX) given as the slope of the friction vs. normal force curve. For two data points $(F_{N,1}, F_{f,1}), (F_{N,2}, F_{f,2})$, $F_{N,1} < F_{N,2}$ we would evaluate the associated friction coefficient $\mu_{1,2}$ as

$$\mu_{1,2} = \frac{F_{f,2} - F_{f,1}}{F_{N,2} - F_{N,1}} = \frac{\Delta F_f}{\Delta F_N}$$

In the pilot study it became clear that the effects on friction under the change of F_N is negligible in comparison to the effects under change of S . Thus, by working under the assumption $F(F_N, S) \sim F(S)$ and a coupling

$F_N \propto R \cdot S$ with coupling ratio R we get

$$\mu_{1,2}(S_1, S_2) = \frac{\Delta F_f(S_1, S_2)}{R(S_2 - S_1)} \propto \frac{\Delta F_f(S_1, S_2)}{\Delta S}, \quad (3.1)$$

With the above reasoning we have in practice exchanged F_N with S in the expression for the friction coefficient. This means that we are interested in a negative slope on the friction vs. stretch curve which corresponds to a negative friction coefficient in our proposed coupled system. The next question remaining is then how to evaluate the strength of this property. By definition, the minimum slope value would give the lowest friction coefficient. However, for two data points with a small ΔS , corresponding to small denominator in Eq. (3.1), would potentially result in big $|\mu|$ without any significant decrease in friction. Hence, we choose to consider the drop in friction with increasing stretch. For a discrete dataset we can locate all local maxima and evaluate the difference to all succeeding local minima. The biggest drop will serve as our indicator for a significant negative friction coefficient. In this evaluation we do not guarantee a monotonic decrease of friction in the range of the biggest drop, but when searching among multiple configurations this is considered a descent strategy to highlight configurations of interest worthy of further investigation.

In addition to the biggest drop in friction we also look at minimum and maximum friction along with the difference between these extrema. In Table 3.2 we summarized the extrema of these properties. The corresponding friction vs. stretch profiles and configurations are visualized for each property category in Fig. 3.3 to 3.6. The stretch profiles for all the configurations are shown in appendix ??.

Table 3.2: Evaluation of the properties of interest for our dataset.

Tetrahedron	Configuration	Stretch	Value [nN]
Min F_{fric}	(3, 9, 4)	0.0296	0.0067
Max F_{fric}	(5, 3, 1)	0.1391	1.5875
Max ΔF_{fric}	(5, 3, 1)	[0.0239, 0.1391]	1.5529
Max drop	(5, 3, 1)	[0.1391, 0.1999]	0.8841

Honeycomb	Configuration	Stretch	Value [nN]
Min F_{fric}	(2, 5, 1, 1)	0.0267	0.0177
Max F_{fric}	(2, 1, 1, 1)	1.0654	2.8903
Max ΔF_{fric}	(2, 1, 5, 3)	[0.0856, 1.4760]	2.0234
Max drop	(2, 3, 3, 3)	[0.5410, 1.0100]	1.2785

Random walk	Configuration	Stretch	Value [nN]
Min F_{fric}	12	0.0562	0.0024
Max F_{fric}	96	0.2375	0.5758
Max ΔF_{fric}	96	[0.0364, 0.2375]	0.5448
Max drop	01	[0.0592, 0.1127]	0.1818

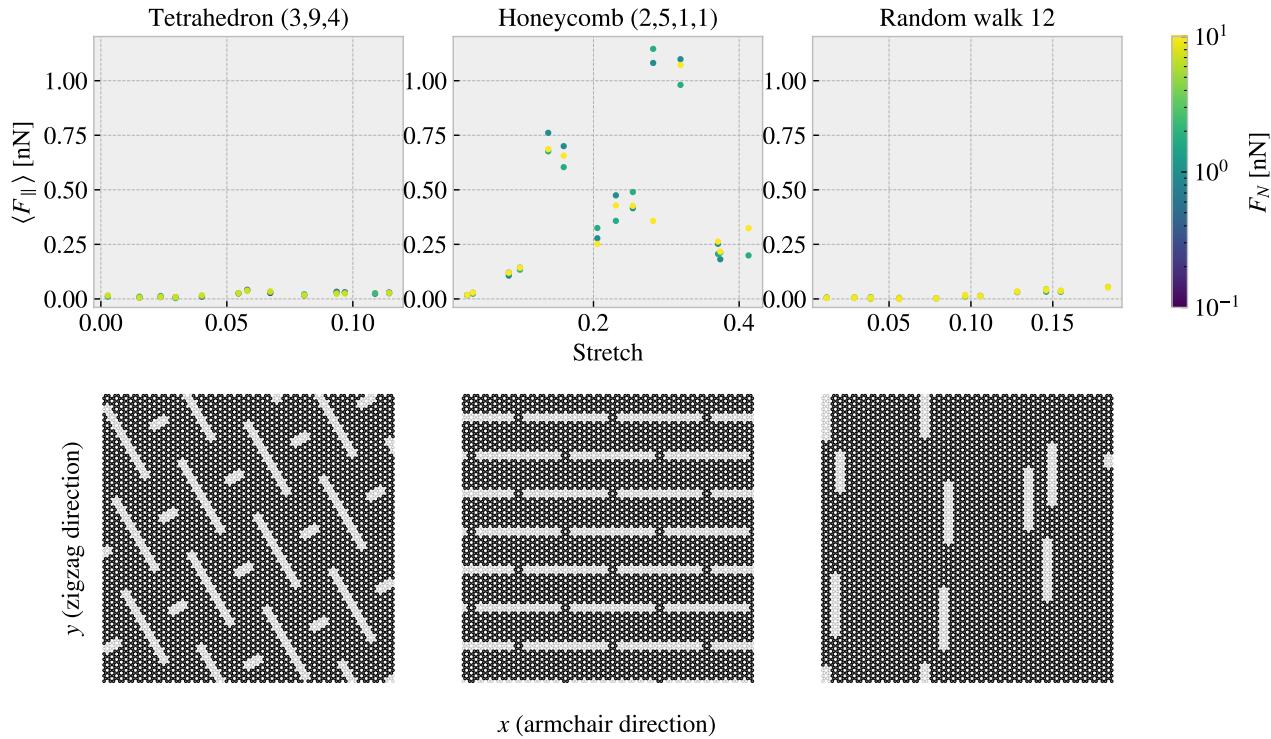


Figure 3.3: Minimum friction: Configurations corresponding to the minimum friction.

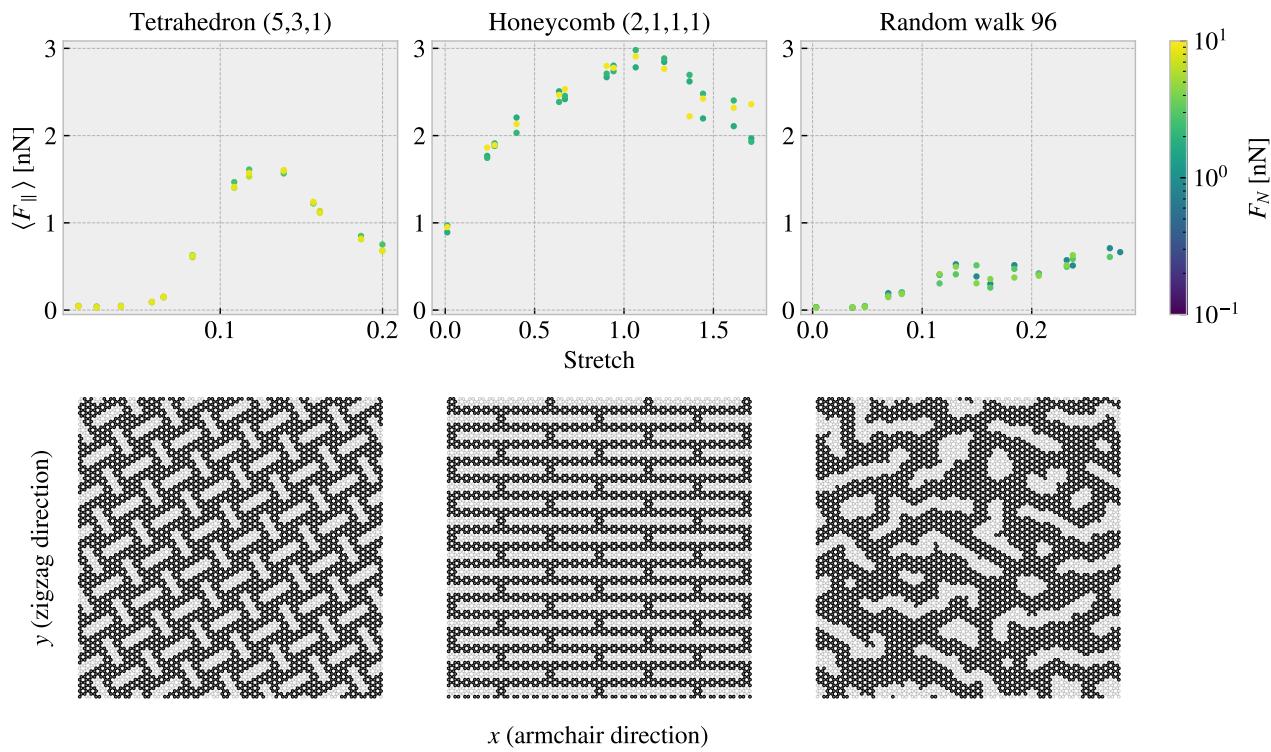


Figure 3.4: Maximum friction: Configurations corresponding to the maximum friction.

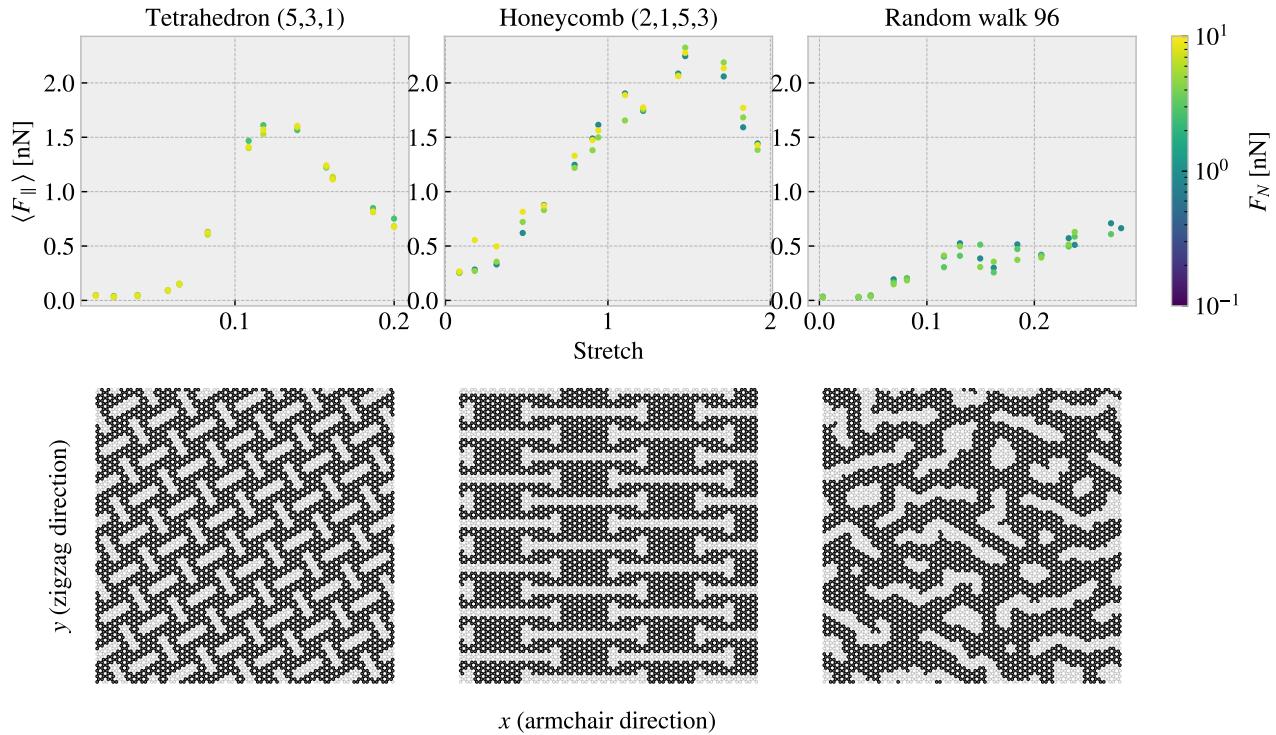


Figure 3.5: Maximum Difference: Configurations corresponding to the biggest difference in friction in the dataset for each pattern.

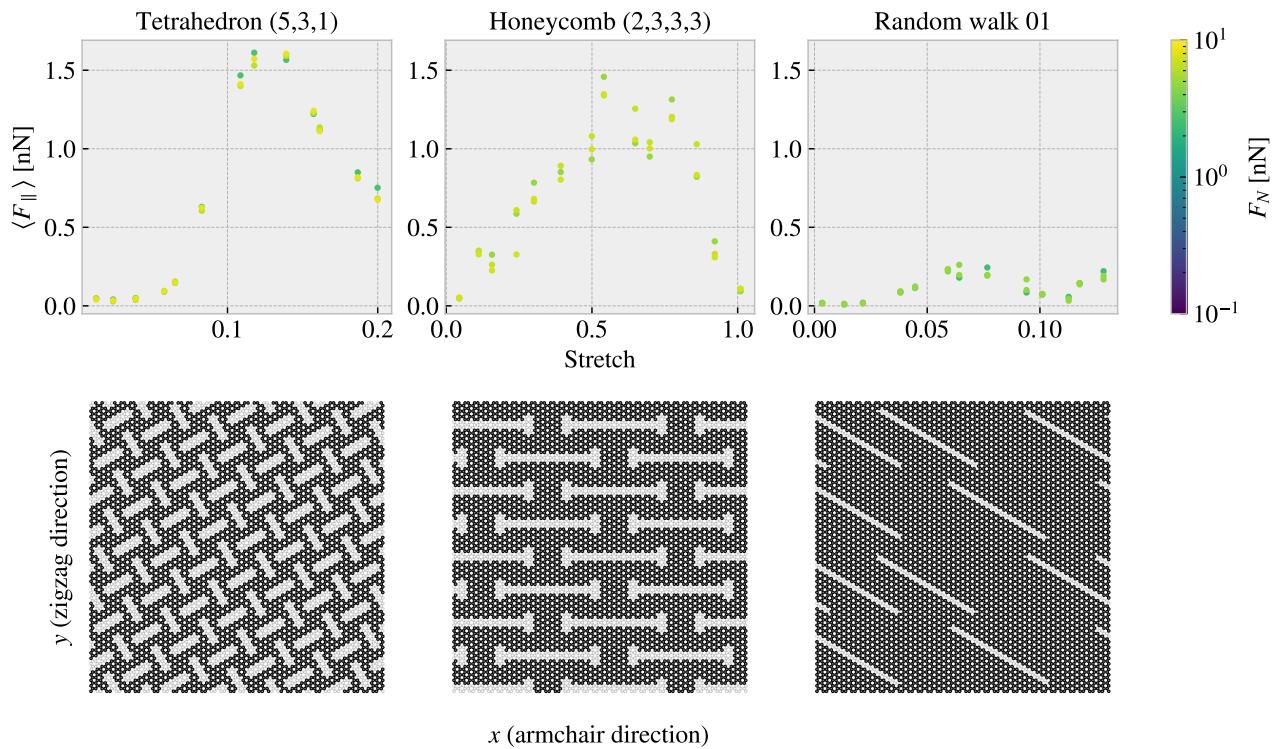


Figure 3.6: Maximum drop: Configurations corresponding to the biggest friction drop in the dataset for each pattern.

3.4 Machine learning

3.4.1 Architecture

Due to the spatial dependencies in the kirigami configurations we use a convolutional neural network (CNN). Studies on similar a similar system envolving the graphene sheet 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. 3.7 shows the key features;

- The image is processed through a series of 3×3 convolutional filters (the smallest size to capture spatial dependencies) using a stride of 1 with an increasing number of channels throughout the network. Each convolutional layer is followed by a ReLU acitivation.
- The spatial dimensions are reduced by a max pooling (2×2 , stride of 2), which half the spatial resolution each time.
- The latter part of the network consist of fully connected part followed by a ReLU activation. The image is first processed in a 1×1 which performs a linear mapping to a series of fully connected layers.

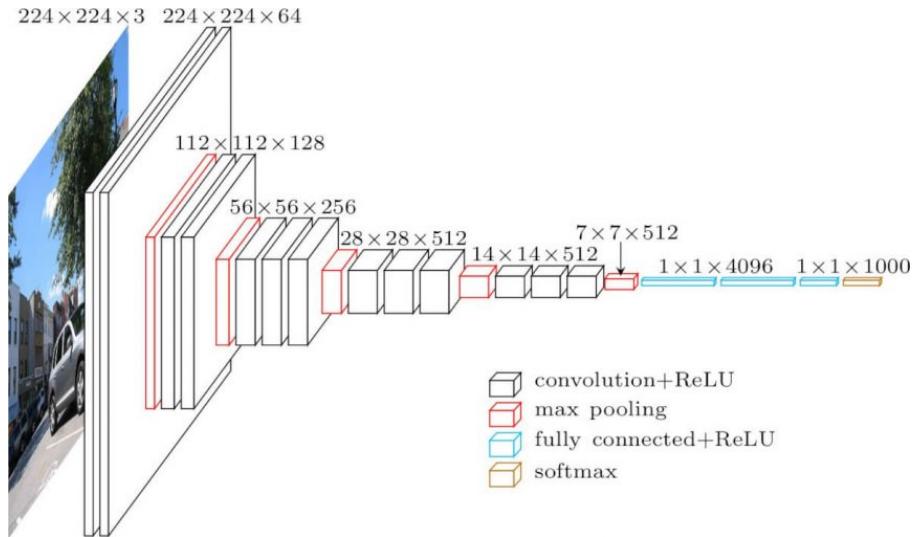


Figure 3.7: VGGNet 16. Source <https://neurohive.io/en/popular-networks/vgg16/>.

In contrast to the VGGNet-16 we restrict ourselves to building the convolutional part in terms of blocks of (convolution, ReLU, max pooling), thus not allowing for any consecutive convolutional filters without performing a max pooling as well. The fully connected blocks is defined as (fully connected, ReLU) similar to that of the VGGNet model. Hanakata et al. and Wan et al. used a similar construction before settling on the models

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

Where C denotes a convolutional block with the following number being the number of channels and D a fully conneted (dense) layer with the number denoting number of nodes. For the process of determing a suiting complexity for the architecture we adpot the approach by Wan et al. [8] who used a “staircase” pattern for combinning convolutional and fully connected blocks. By defining a starting number of channels S and network depth D we fill the first half with convoliutional blocks doubling in channel number for each layer and the latter half with fully connected blocks starting setting the number of nodes as the reverse pattern used for the number of channels. Following this pattern a ($S = 4, D = 8$) would take the form

$$\text{Input} \rightarrow \overbrace{C4 \quad C8 \quad C16 \quad C32 \quad D32 \quad D16 \quad D8 \quad D4}^{S=4 \atop D=8} \rightarrow \text{Output.}$$

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

3.4.2 Data handling

3.4.2.1 Input

We use three variables as input: Kirigami configuration, stretch of the sheet and applied normal load. While the first is a two-dimensional input the latter are both scalar values. This gives rise to two main options for the data structure

1. Expand the scalar values (stretch and load) into 2D matrices of the same size as the kirigami configuration by copying the scalar value to all positions. This can then be merged into an image of three channels used as a single input.
2. Pass only the kirigami configuration through the CNN part of the network and introduce the remaining scalar values into the FC part of the network.

Both options utilize the same data, but the first emphasizes that the configurations should be processed in relation to the applied stretch and load, while the latter represent a more independent processing. We implemented the option to do both variations, but it quickly became clear that option 1 was producing the most promising result (hldo more rigorous presentation of this?).

3.4.2.2 Output

For the output we are mainly concerned about the mean friction and the rupture detection. In combination this let us produce friction vs. stretch curves with the rupture detection used to predict the stopping point of the curve in terms of a maximum stretch value. However, it has often been proven useful to introduce more variables in the output in order to strengthen the network ([get source](#)). In addition, this gives us more option for exploring the relationship in the data later on. Thus, we include maximum friction, contact count, porosity and rupture stretch in the data. Notice that rupture stretch refers to the value found in the rupture test without load, but as the sheet always ruptures before or just around this point in a loaded state this provides some information for the training to lean on, even though it is in the output state. In principle we could add a penalty whenever the network predicts the sheet to be attached for stretch values above the rupture stretch, but we found the performance of the rupture prediction to be satisfactory without introducing such formalities. Notice that we weight the importance of these variables differently as explained in the section about the loss function.

3.4.2.3 Data augmentation

Due to the symmetry in the configuration, we can flip the configuration by reflecting the configuration across the y-axis. We cannot do this across the x-axis as the sheet is always slide in a positive y-direction and thus we cannot guarantee that the results would be the same whether we slide backwards or forward. We definitely expect a snow plow to perform differently when going backwards and thus by analogy we would expect the direction of sliding with respect to the configuration to be of importance.

3.4.3 Loss

We use a rather standard loss function for the evaluation of our model performance during training and validation. For the scalar values we use the Mean Squared Error (MSE)

$$L_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2,$$

where N is the number of data entries and y are the true variables and \hat{y} are the predicted values. For the prediction of rupture with True/False labels we use binary cross entropy

$$L_{\text{BCE}} = -\frac{1}{N} \left[\sum_{i=1}^N [t_i \log(p_i) + (1 - t_i) \log(1 - p_i)] \right],$$

where $t \in \{0, 1\}$ is the truth label. Does this belong in theory entirely?

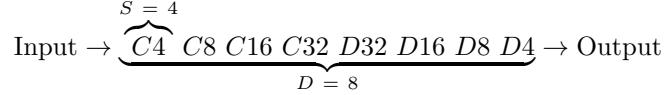
We calculate the total loss as a weighted sum between the loss associated with each variable

$$L_{tot} = \sum_v W_v \cdot L_v.$$

We choose the weights to be $1/2$ for the mean friction and $1/10$ for the remaining 5 variables thus sharing weight evenly for the remaining 50% of the loss. During the introductory phase of the training we tried varying these, but we found little change in the results and concluded immediately that the precise choice of this weighting is not immensely important.

3.4.4 Hypertuning

We build a series of models by VGGNet type. The first part consist of convolutional layers and then final part of fully connected layers. Each convolutional layer is followed by a Batchnorm, then a Relu and finally a maxpool. As we go deeper into the network we increase the number of channels in the convolutional layers, and similarly we decrease the number of nodes in the fully connected layers towards the end. We choose an even distribution of C and FC layers. We do so in a staircase kind of way such, as for instance



By varying the number of start channels S and the depth D we perform a architecture search. However, the appropriate learning rate will depend on the specific architecture. Hence, we perform a learning rate range test for each of the suggested model architectures before doing the search. This is shown in figure Fig. 3.8.

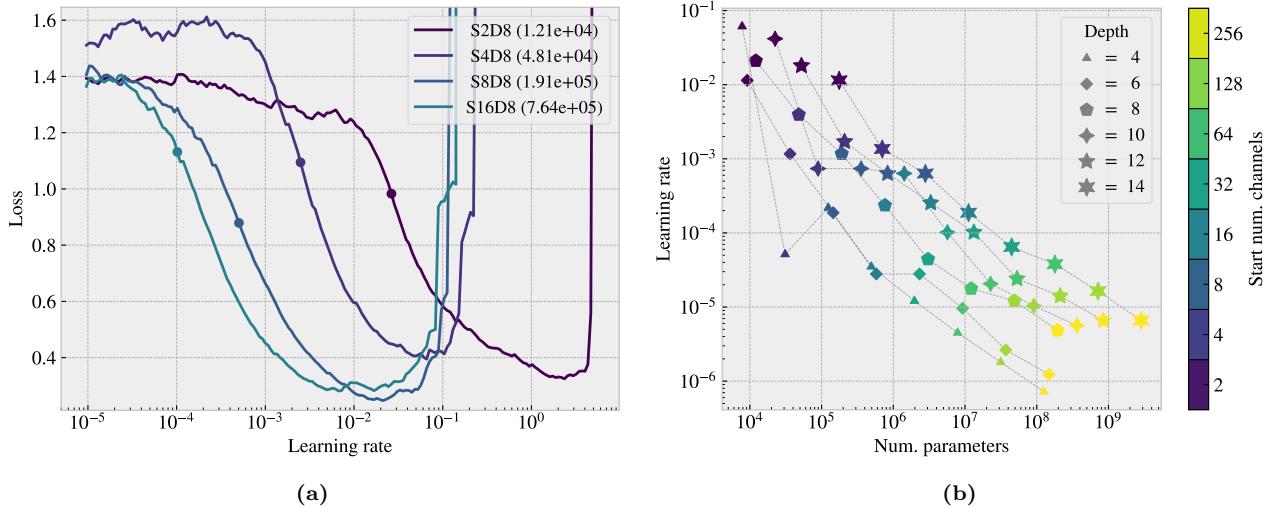


Figure 3.8: Learning rate range test. (a) show a few examples of how the test ran. (b) show the complete result from the test.

We then use these suggested learning rates for the architecture search. We look at the total validation loss and the validation R2 score for the mean friction. This is shown in Fig. 3.9

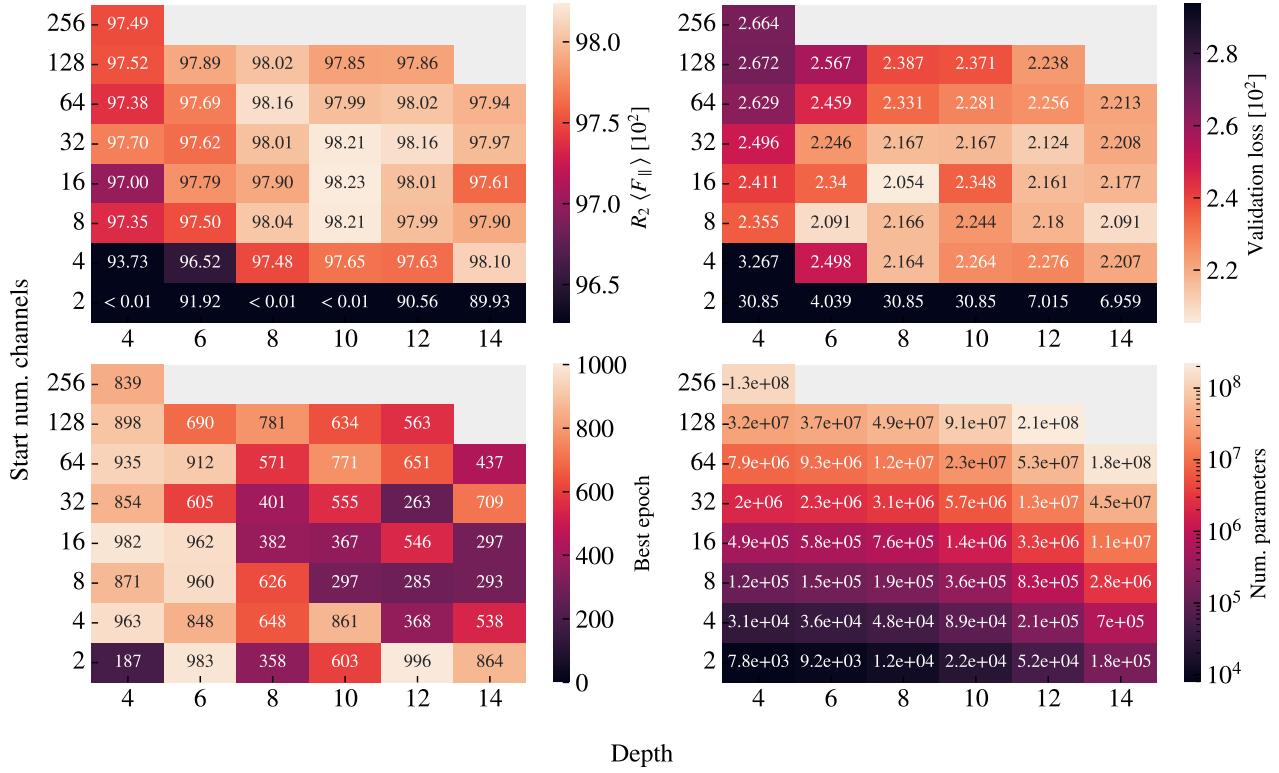


Figure 3.9: Architecture search

Now look at the result for the more difficult patterns as a pseudo validation (these are actually expected to be partly in the training set already).

We look at the honeycomb and tetrahedron patterns with top 10 scores for maximum drop in friction. This is chosen as an attempt to capture the ones with the most complex non-linear trends in the stretch curve which seems to be both difficult to predict but also of most interest for the study. This is shown in Fig. 3.10.

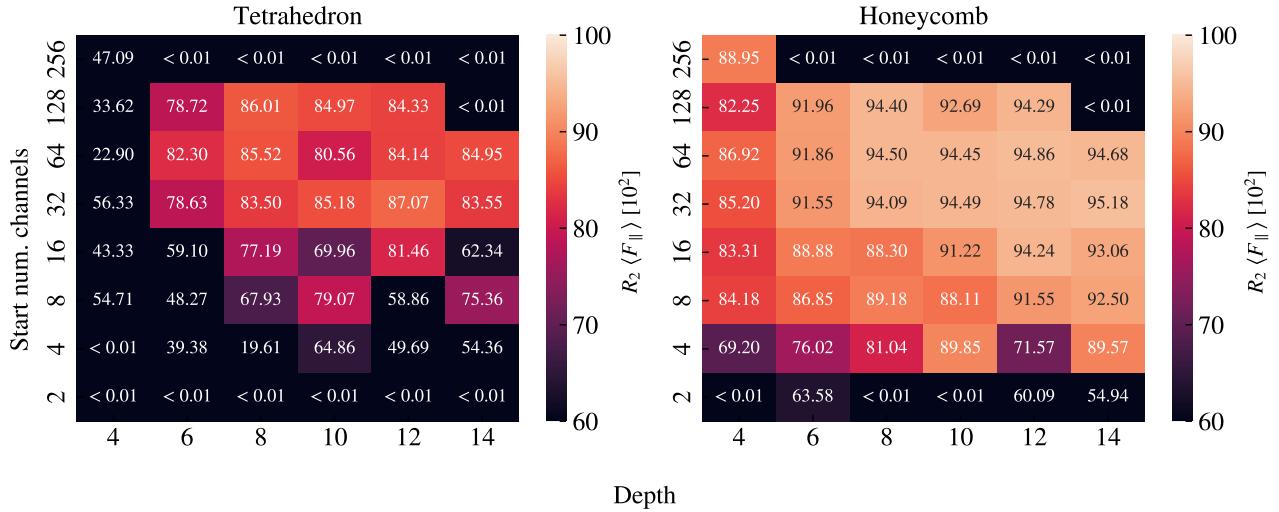


Figure 3.10: Selected pseudo validation set

From Fig. 3.9 we found that models like S(8-32)D(8-12) gave reasonable results. When looking at the selected validation set in figure Fig. 3.10 we get considerably lower scores, especially for the Tetrahedron pattern. Here

the best performing models were in general more complex with S32D12 being the best candidate for the tetrahedron and S32D12 the best for the honeycomb pattern. This is compatible with the general performance from Fig. 3.9 and by favoring the performance on the tetrahedron set (which seemed to be more difficult) we settle on the S32D12 model.

We perform a lr range test with different momentum at this time we find the maximum lr at the point where the loss diverges as shown in figure Fig. 3.11.

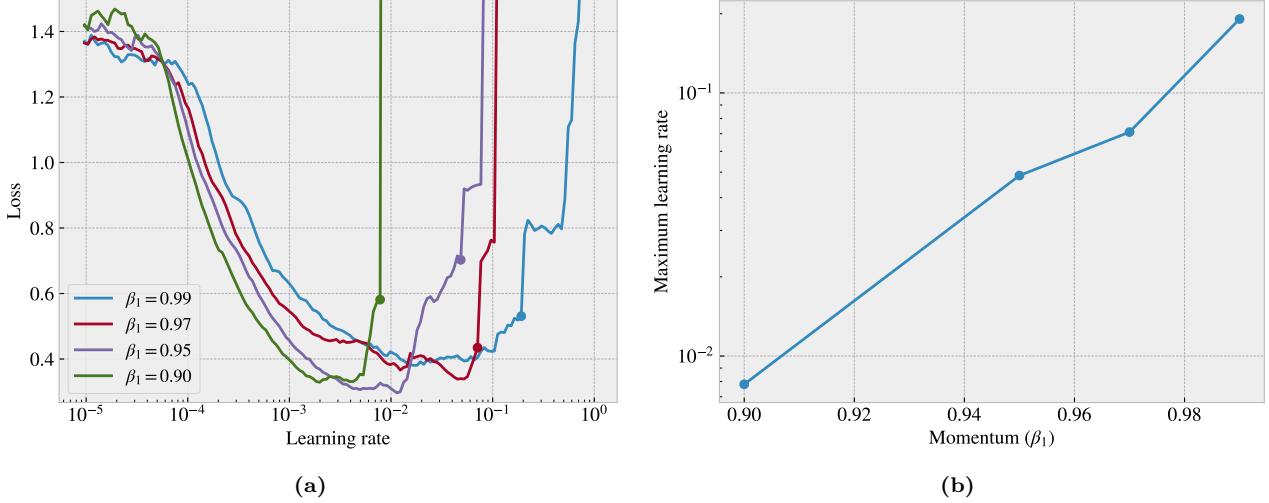


Figure 3.11: ...

The LR range is done around 200 steps (before it diverges). So if I figure out the number of batches I can check how many epochs is default since number of steps is steps per epoch times number of epochs.

Seems like there are 242 batches in the full dataset so it probably just do one epoch...

Choices to make

- Learning rate: increasing LR to locate minimum in loss gradient (fastest learning) (LR range test) cycling learning rates maybe?
- Architecture: VGGNet staircase type search
- Optimizers: Just try some different ones and look at convergence
- weight decay

Suggestion for hyperparameter tuning

1. Select complexity range of VGGNet staircase type architectures
2. Start with Adam optimizer at default settings
3. Perform LR range test to get information of a suitable LR and choose one that will work for most of the architectures.
4. Perform architecture grid search over depth and start number of channels (complexity). Choose best architecture from this.
5. Additionally add weight decay to further optimize learning for that architecture choice. Perhaps cyclic learning rates.

Start by mentioning the related articles using a VGGNet type network (and maybe also their learning rates).

Either submit with best lr for each or choose something that works for all. 0.0005 seemed as a good middle ground but the complex networks crash with these learning rates. It looks like 0.0001 (like Hanakata used) will do the job for all.

3.5 Accelerated Search

Having a network model that can predict friction force for a given configuration are able to search for some desired properties. Low and high friction and maximal negative friction coefficients

Here we pursue two different approaches for finding

1. Generate an enlarged dataset and run it through the ML model
2. Genetic algorithm

3.5.1 Markov-Chain Accelerated Genetic Algorithms

3.5.1.1 Talk about traditional method also?

3.5.1.2 Implementing for 1D chromosome (following article closely)

We have the binary population matrix $A(t)$ at time (generation) t consisting of N rows denoting chromosomes and with L columns denoting the so-called locus (fixed position on a chromosome where a particular gene or genetic marker is located, wiki). We sort the matrix rowwise by the fitness of each chromosome evaluated by a fitness function f such that $f_i(t) \leq f_k(t)$ for $i \geq k$. We assume that there are a transition probability between the current state $A(t)$ and the next state $A(t+1)$. We consider this transition probability only to take into account mutation process (mutation only updating scheme). During each generation chromosomes are sorted from most to least fitted. The chromosome at the i -th fitted place is assigned a row mutation probability $a_i(t)$ by some monotonic increasing function. This is taken to be

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

for some limit N' (refer to first part of article talking about this). We use $N' = N/2$. We also define the survival probability $s_i = 1 - a_i$. In thus way a_i and s_i decide together whether to mutate to the other state (flip binary)

or to remain in the current state. We use s_i as the statistical weight for the i -th chromosome given it a weight $w_i = s_i$.

Now the column mutation. For each locus j we define the count of 0's and 1's as $C_0(j)$ and $C_1(j)$ respectively. These are normalized as

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

These are gathered into the vector $\mathbf{n}(j, t) = (n_0(j, t), n_1(j, t))$ which characterizes the state distribution of j -th locus. In order to direct the current population to a preferred state for locus j we look at the highest weight of row i for locus j taking the value 0 and 1 respectively.

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

which is normalized again

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

The vector $\mathbf{n}(j, t+1) = (n_0(j, t+1), n_1(j, t+1))$ then provides a direction for the population to evolve against. This characterizes the target state distribution of the locus j among all the chromosomes in the next generation. We have

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

Since the probability must sum to one for the rows in the P-matrix we have

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

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

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

We just need to know $P_{00}(j, t)$. We start from $P_{00}(j, t=0) = 0.5$ and then choose $P_{00}(j, t) = n_0(j, t)$

Appendices

Appendix A

Appendix B

Appendix C

Bibliography

- ¹E. Gnecco and E. Meyer, *Elements of friction theory and nanotribology* (Cambridge University Press, 2015).
- ²Bhusnur, “Introduction”, in *Introduction to tribology* (John Wiley & Sons, Ltd, 2013) Chap. 1, 1–?
- ³H.-J. Kim and D.-E. Kim, “Nano-scale friction: a review”, *International Journal of Precision Engineering and Manufacturing* **10**, 141–151 (2009).
- ⁴K. Holmberg and A. Erdemir, “Influence of tribology on global energy consumption, costs and emissions”, *Friction* **5**, 263–284 (2017).
- ⁵B. 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.
- ⁶P. 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).
- ⁷P. 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).
- ⁸L.-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).
- ⁹Y. Mao, Q. He, and X. Zhao, “Designing complex architectured materials with generative adversarial networks”, *Science Advances* **6**, eaaz4169 (2020).
- ¹⁰Z. 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).
- ¹¹A. 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).
- ¹²S. 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).
- ¹³Z. 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).
- ¹⁴R. W. Liefferink, B. Weber, C. Coulais, and D. Bonn, “Geometric control of sliding friction”, *Extreme Mechanics Letters* **49**, 101475 (2021).