

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

Designs for 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

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

Designs for a negative friction coefficient.

Mikkel Metzsch Jensen



© 2023 Mikkel Metzsch Jensen

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

<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. 12

LJ Lennard-Jones. 37

MD Molecular Dynamics. 5, 12, 19, 23, 26, 31, 35, 36

ML Machine Learning. 20, 21, 22, 23, 24, 25, 37

MSE Mean Squared Error. 13

Contents

| | |
|---|-----------|
| I Background Theory | 1 |
| II Simulations | 3 |
| 1 Kirigami configuration exploration | 5 |
| 1.1 Generating the dataset | 5 |
| 1.2 Data analysis | 6 |
| 1.3 Properties of interest | 8 |
| 1.4 Machine learning | 12 |
| 1.4.1 Architecture | 12 |
| 1.4.2 Data handling | 13 |
| 1.4.2.1 Input | 13 |
| 1.4.2.2 Output | 13 |
| 1.4.2.3 Data augmentation | 13 |
| 1.4.3 Loss | 13 |
| 1.4.4 Hypertuning | 14 |
| 1.4.5 Final model | 19 |
| 1.5 Accelerated Search | 23 |
| 1.5.1 Patteren generation search | 23 |
| 1.5.2 Genetic algorithm search | 27 |
| 2 Negative friction coefficient | 31 |
| 2.1 Nanomachine coupling | 31 |
| 3 Summary | 35 |
| 3.1 Summary and conclusions | 35 |
| 3.1.1 Design MD simulations | 35 |
| 3.1.2 Design Kirigami framework | 35 |
| 3.1.3 Control friction using Kirigami | 35 |
| 3.1.4 Capture trends with ML | 36 |
| 3.1.5 Accelerated search | 37 |
| 3.1.6 Negative friction coefficient | 37 |
| 3.2 Outlook / Perspective | 37 |
| Appendices | 39 |
| A Appendix A | 41 |
| A Appendix B | 43 |
| B Appendix C | 45 |

Part I

Background Theory

Part II

Simulations

Chapter 1

Kirigami configuration exploration

Building upon the discoveries of the Pilot Study ??, 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 towards 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 gain a broader understanding of the friction-strain relationship. We will then leverage this dataset to explore the potential of employing machine learning for predicting friction behavior based on Kirigami design, strain, and load. Finally, we plan to utilize the developed machine learning model for an accelerated search.

1.1 Generating the dataset

We aim to create a dataset that contains an extended series of Kirigami design configurations based on the pattern generation methods developed in ?? for which we will vary the strain and load for each configuration. For each configuration, we sample 15 pseudo uniform strain values (see ??) between zero and the rupture strain according to the rupture test. Since the normal force did not prove to be dominant in the friction description we only sample 3 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 values presented in the pilot study (see ??). 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 that results in translation of the patterns. A summary of the dataset is given in Table 1.1 while all configurations are shown visually in ???. The Tetrahedron and Honeycomb parameters are chosen to provide additional variations of the configurations evaluated in ?? which exhibited interesting properties. The Random walk configurations are chosen with the aim of introducing as much variety as possible within our Random walk framework. Notice that not all submitted data points “make it” to the final dataset. This is due to a small bug in the data generation procedure¹.

¹The issue arises from the fact that the rupture point in the rupture test does not completely 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 (LAMMPS 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 checkpoint 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 quite have been mitigated by a rewrite of the code, but it was first discovered after the dataset had been created. We notice, however, that the dataset still contains 11.57 % rupture events which provide a reasonable amount of rupture events to incorporate in the machine learning model

Table 1.1: 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 “make it” to 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. In the latter datasets, the reference point for the pattern 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 %) |

1.2 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

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

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. 1.1.

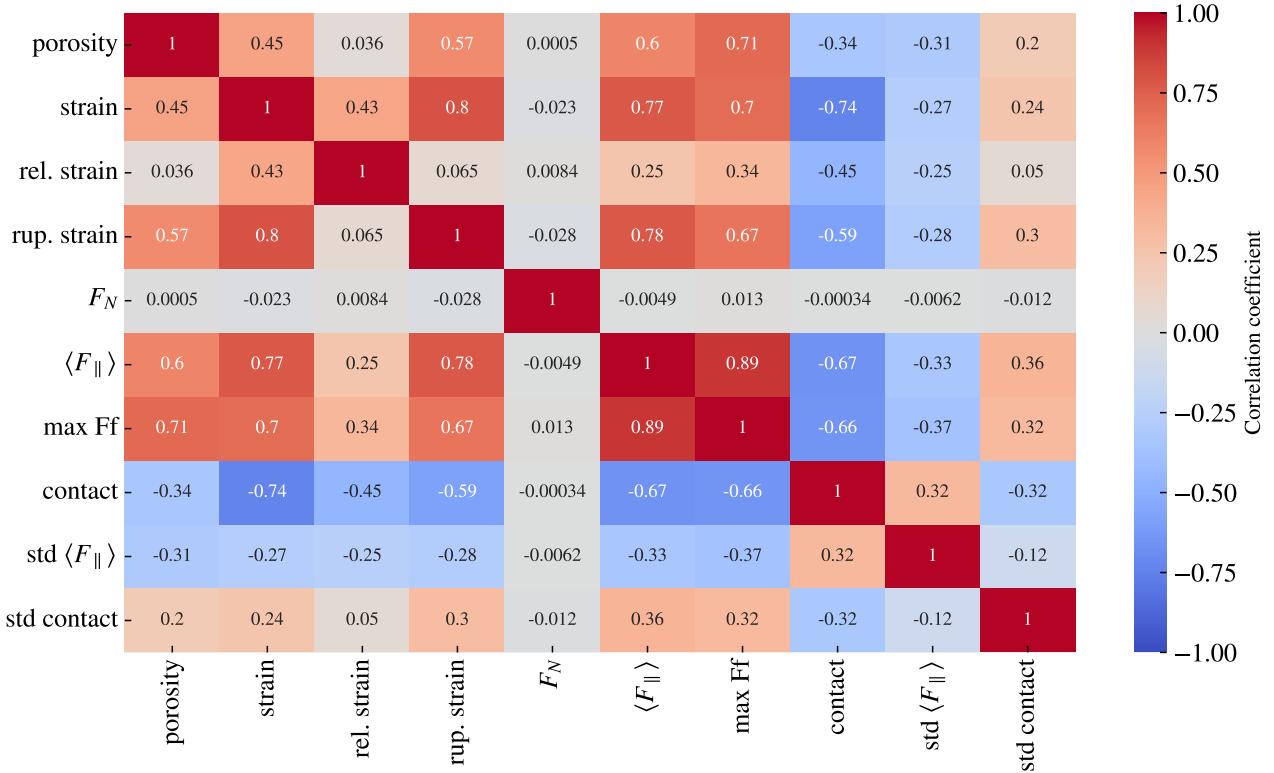


Figure 1.1: Pearson product-moment correlation coefficients for the full dataset (see Table 1.1).

From Fig. 1.1 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, the 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). From Fig. 1.1 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 ?? where the increasing strain was correlated 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 (??) 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. 1.2 shows a visualization of the data (excluding the pilot study configurations) for chosen variable pairs on the axes. This provides a visual clue on some of the correlations and provides a qualitative feeling for the diversity in various planes of the feature space that we eventually will base our machine learning model on. We notice that the honeycomb pattern is spanning a significantly larger range of strain, contact and mean friction.

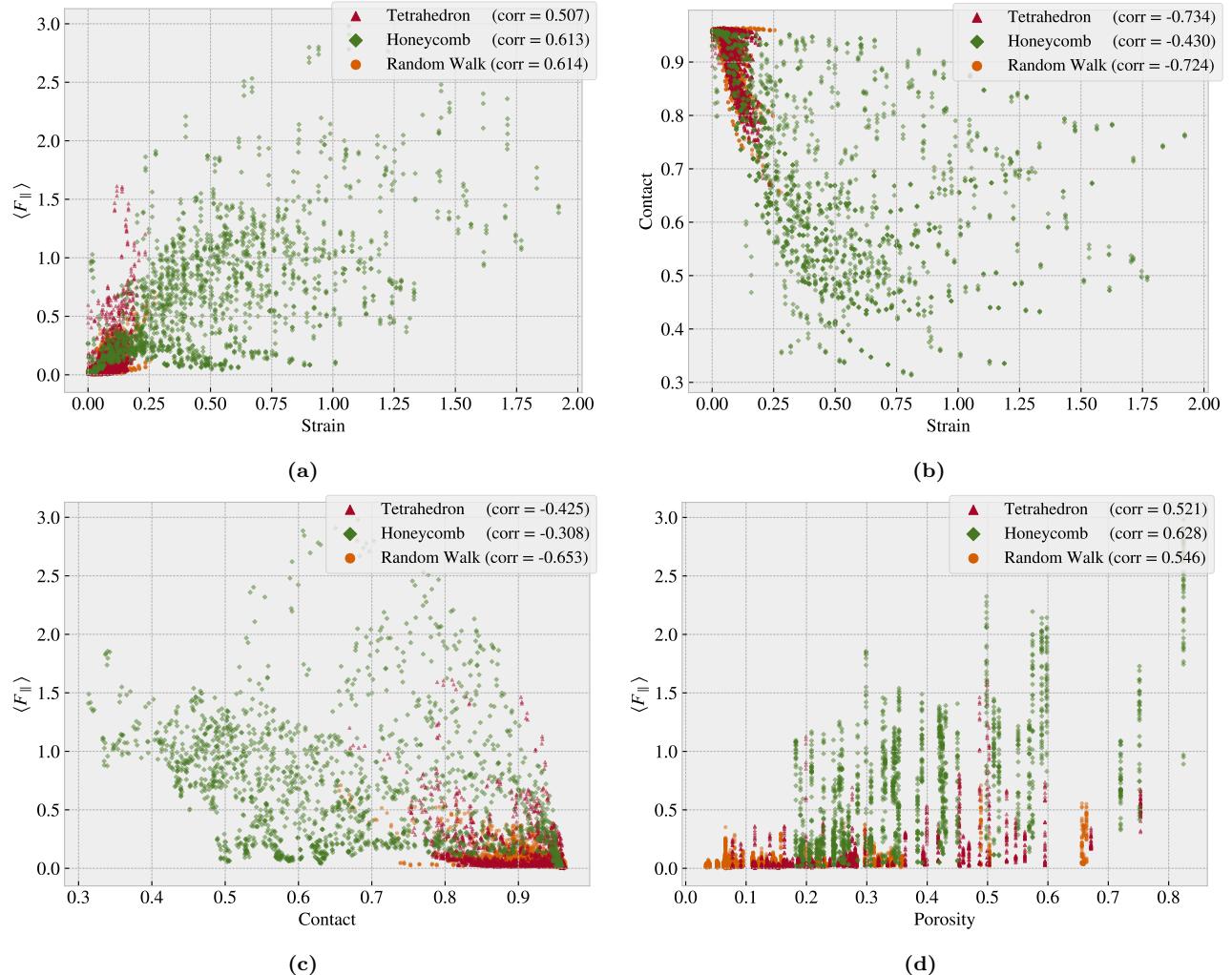


Figure 1.2: Scatter plot of the data sets Tetrahedron, Honeycomb and Random Walk (excluding the pilot study) for various variable combinations in order to visualize some chosen correlations of interest and distributions in the data

1.3 Properties of interest

In the pilot study (??) we found promising results for the idea of achieving a negative friction coefficient under the assumption of a system with coupled normal force and strain. Hence, we will consider this as a main property of interest for our further exploration. However, it is not obvious how one should rigorously quantify this. The friction coefficient is by our definition (??) given as the slope of the friction F_f vs. normal force F_N curve. Hence, for two data points $\{(F_{N,1}, F_{f,1}), (F_{N,2}, F_{f,2})\}$, $F_{N,1} < F_{N,2}$ we can 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 apparent that the effects of friction under the change of load is negligible in comparison to the effects related to strain ε . Thus, by working under the assumption $F(F_N, \varepsilon) \sim F(\varepsilon)$ and a coupling $F_N \propto R \cdot \varepsilon$ with linear coupling ratio R we get

$$\mu_{1,2}(\varepsilon_1, \varepsilon_2) = \frac{\Delta F_f(\varepsilon_1, \varepsilon_2)}{R(\varepsilon_2 - \varepsilon_1)} \propto \frac{\Delta F_f(\varepsilon_1, \varepsilon_2)}{\Delta \varepsilon}. \quad (1.1)$$

With this reasoning, we can in practice substitute load F_N for strain ε in the expression for the friction coefficient of our coupled system. 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. (1.1), would potentially lead to a huge negative slope value without any significant decrease in friction. Hence, we choose to consider the drop 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* which serves as our indicator for a 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 decent strategy to highlight configurations of interest worthy of further investigation. In addition to the biggest drop in friction, we also consider the minimum, $\min F_{\text{fric}}$, and maximum, $\max F_{\text{fric}}$, friction along with the maximum difference, $\max \Delta f_{\text{fric}} = \max F_{\text{fric}} - \min F_{\text{fric}}$. The extrema of these four properties for each of the categories: Tetrahedron, Honeycomb, Random walk and Pilot study, are summarized in Table 1.2. The corresponding strain profiles and configurations are shown in Fig. 1.3 to 1.6 (excluding the pilot study). The strain profiles for the full dataset are shown in appendix ??.

From the property comparison in Table 1.2 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 still resulting in the highest scores for the maximum properties while the minimum friction is still achieved by the non-cut sheet. This latter observation reveals that our dataset does not provide any indication that friction can readily be reduced for a Kirigami sheet under strain. However, the improvement in the remaining properties indicates that the dataset contains the necessary information to 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 that of the Tetrahedron and Honeycomb structures. However, since these are found to be on a comparable order of magnitude we argue that contain relevant information for populating configuration space with respect to machine learning training. The Random walk patterns also contribute with some immediate insight into the structures that we can associate with each of the properties of interest due to its increased diversity in comparison to the other patterns.

For the $\min F_{\text{fric}}$ top candidates (Fig. 1.3) we find that the Random walk candidate has a rather cut density (low porosity) and with 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 constant level of contact. For the minimum candidate for the Tetrahedron pattern, we also observe a low decrease in contact, and in both these cases this corresponds with a relatively flat friction-strain curve. When considering the remaining friction-strain curve throughout Fig. 1.3 to 1.6 we find a rising friction-strain curve which is accompanied by a decreasing relative contact. When looking at the Random walk 96 pattern, which is the top candidate for both the $\max F_{\text{fric}}$ (Fig. 1.4) and $\max \Delta f_{\text{fric}}$ (Fig. 1.5) property, we find a rather porous configuration with mainly horizontal-orientated cuts.

This has some structural reminiscence with the Honeycomb pattern. Finally, the best random walk candidate for the max drop category, Random walk 01, did not produce a big drop. We find that the contact area is decreasing with strain but not as strongly as seen for the other configurations. The configuration contains some slanted cuts which might be reminiscent of parts of the Tetrahedron pattern.

Table 1.2: Evaluation of the properties of interest for our dataset.

| Tetrahedron | Configuration | Strain | Value [nN] | Hon. (2, 2, 1, 5) [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] | Tetra. (7, 5, 1) [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 |
| $\max F_{\text{fric}}$ | Honeycomb | 0.7279 | 1.5948 |
| $\max \Delta F_{\text{fric}}$ | Honeycomb | 0.7279 | 1.5325 |
| max drop | Honeycomb | [0.7279, 1.0463] | 0.9674 |

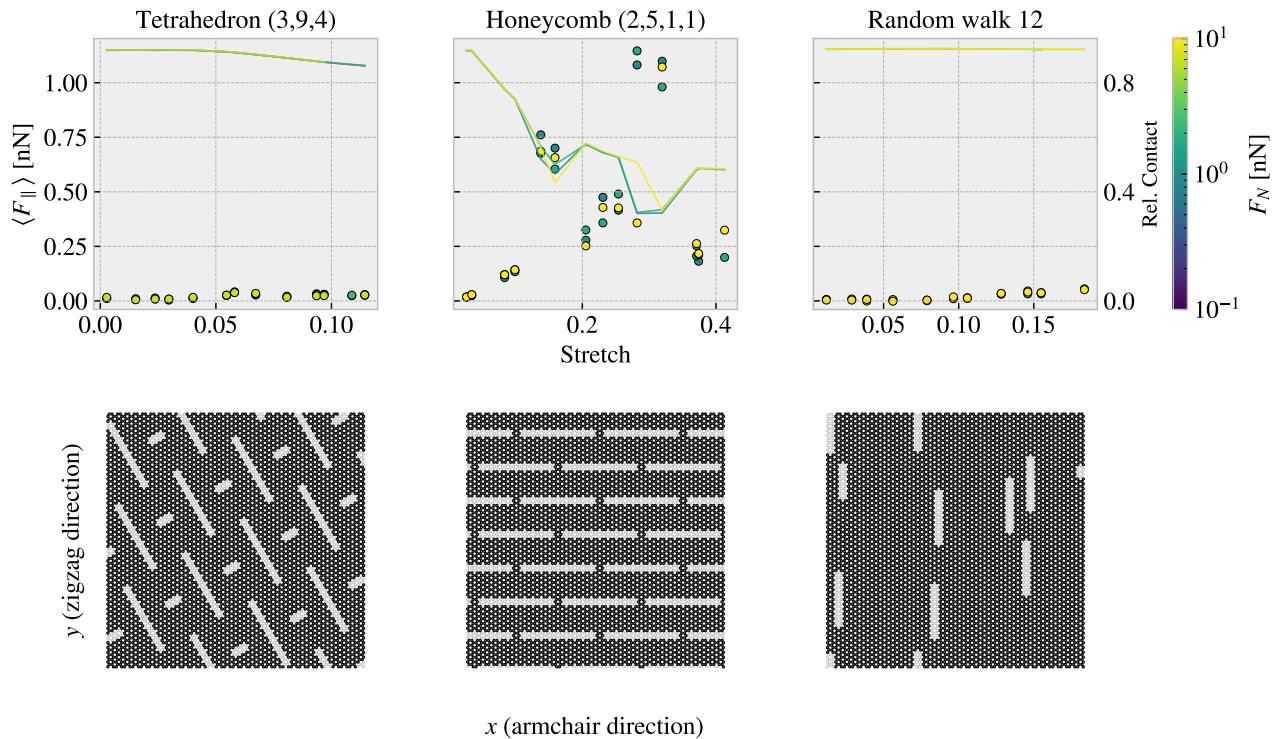


Figure 1.3: Minimum friction: Configurations corresponding to the minimum friction.

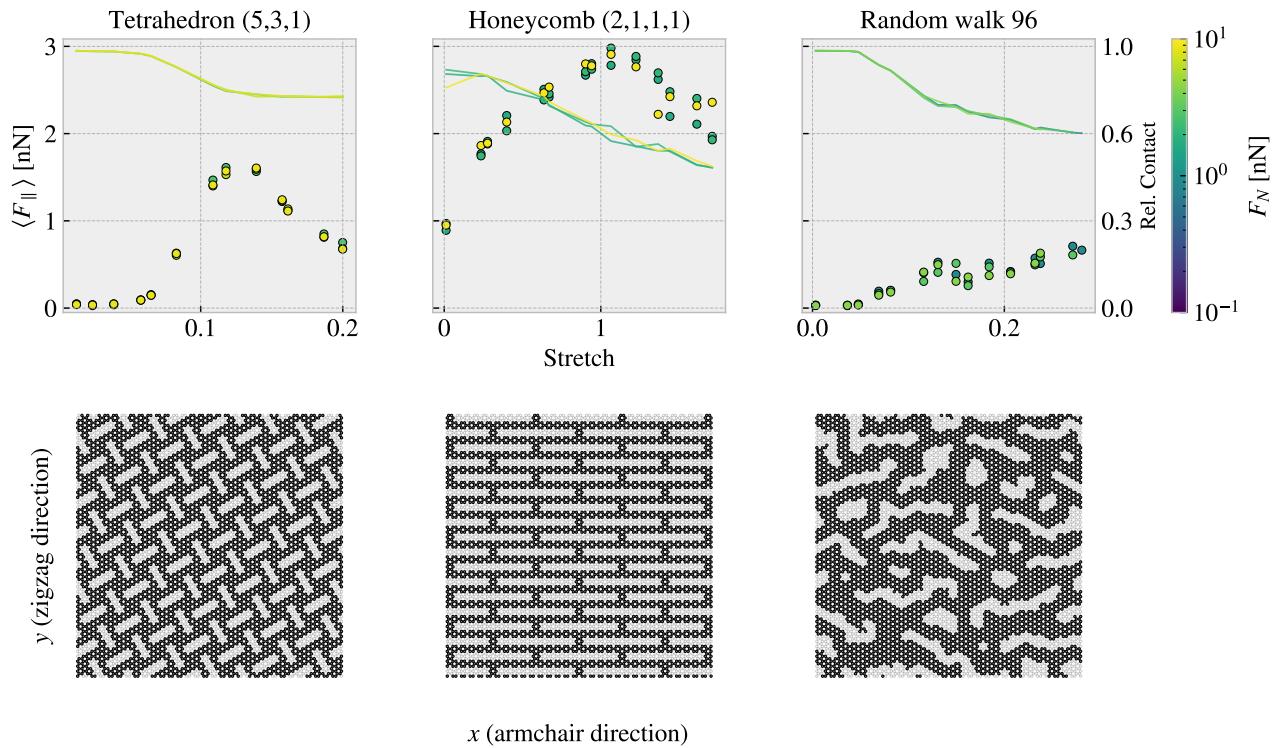


Figure 1.4: Maximum friction: Configurations corresponding to the maximum friction.

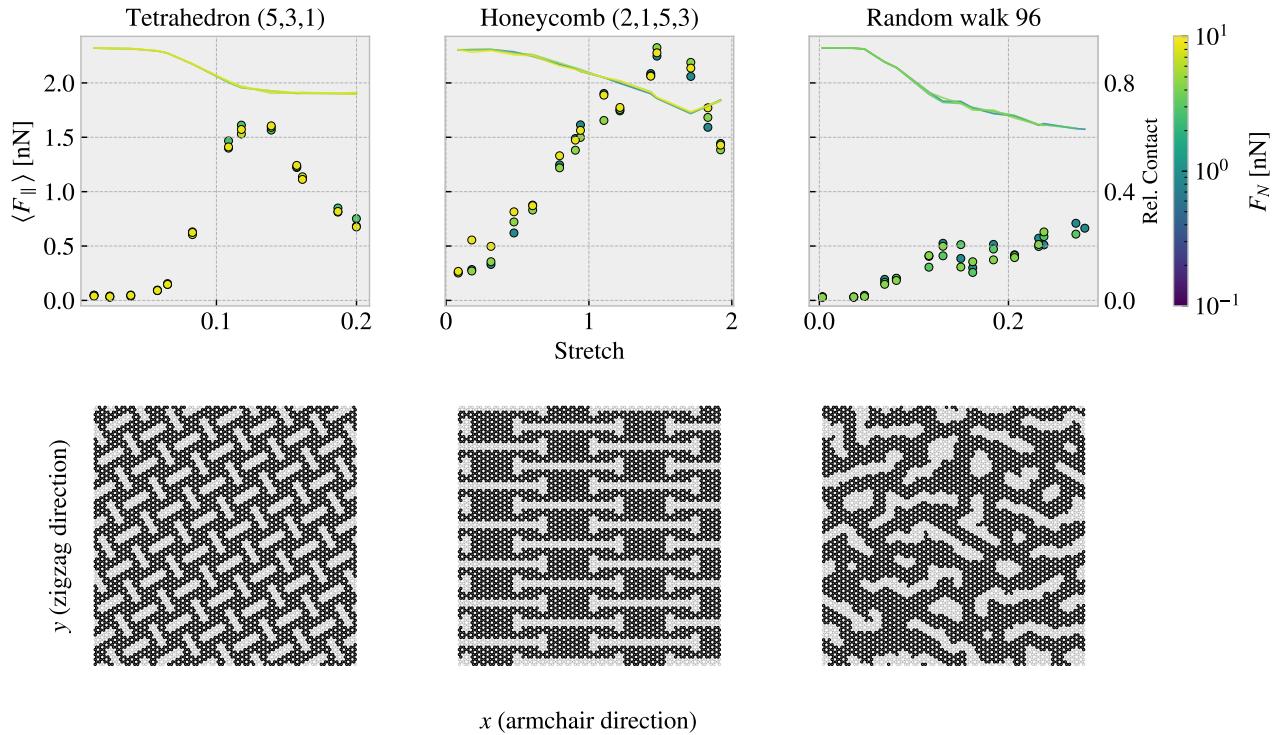


Figure 1.5: Maximum Difference: Configurations corresponding to the biggest difference in friction in the dataset for each pattern.

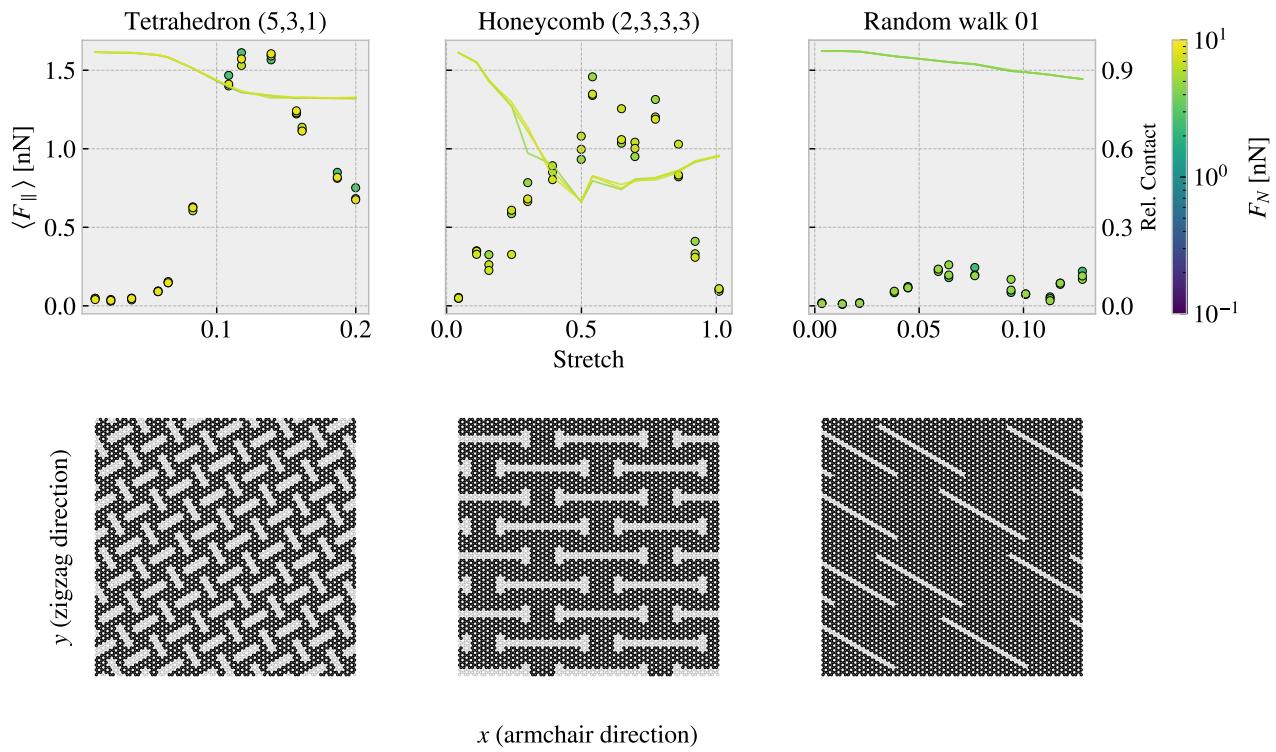


Figure 1.6: Maximum drop: Configurations corresponding to the biggest friction drop in the dataset for each pattern.

1.4 Machine learning

Given the MD-based dataset we investigate the possibilities of training a machine learning model to predict the friction behavior from a given Kirigami configuration, strain and load.

1.4.1 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 network, Hanakata et al. [1, 2] and Wan et al. [3], which we adopt for this study as well. The VGGNet-16 architecture illustrated in Fig. 1.7 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 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, 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 with the number of channels corresponding to the nodes in the first fully connected layer.

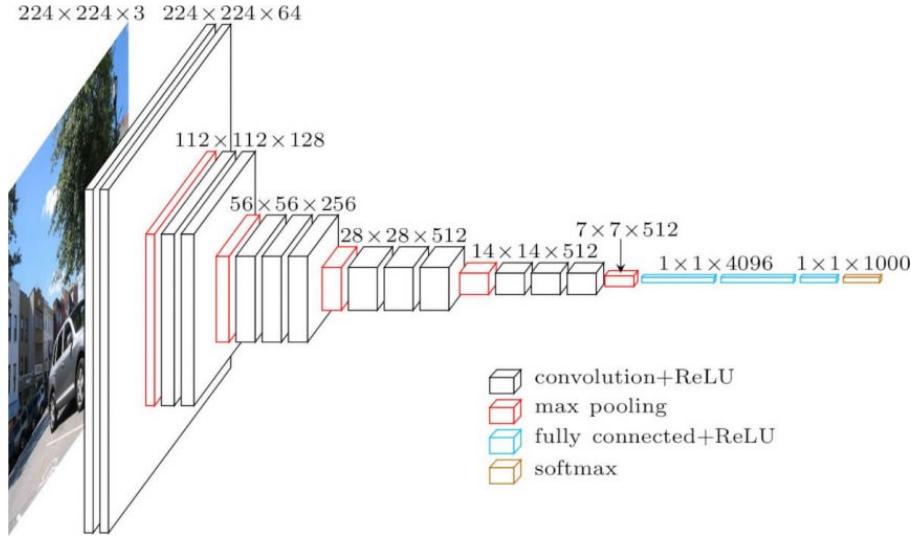


Figure 1.7: VGGNet-16. Source <https://neurohive.io/en/popular-networks/vgg16/>.

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 blocks: (Convolution → Batch normalization → ReLU → Max pooling). Similarly, we define a fully connected block by two elements (Fully connected → ReLU) which match the VGGNet model. Hanakata et al. and Wan et al. used a similar architecture with the parameters

$$\begin{array}{ll} \text{Hanakata et al. [1]} & C16 C32 C64 D64, \\ \text{Wan et al. [3]} & C16 C32 D32 D16, \end{array}$$

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 process of determining a suiting complexity for the architecture, we adopt the approach by Wan et al. [3] who used a “staircase” pattern for combining

convolutional and fully connected blocks. By defining a starting number of channels S and 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 having the number of nodes decrease in a reverse pattern. For instance, the architecture $S4D8$ will take the form

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

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

1.4.2 Data handling

1.4.2.1 Input

We use three variables as input: Kirigami configuration, strain of the sheet and applied normal load. The configuration is given as a two-dimensional input by the 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 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 into the fully connected part of the network halfway in.

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

1.4.2.2 Output

For the output, we are mainly concerned with mean friction and the rupture detection. In combination, this 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. 1.4.3.

1.4.2.3 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, flip 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. We cannot use a reflection across the x-axis as the sheet is sliding in a positive y-direction. This would correspond to a change in the sliding direction which we cannot expect to be fully symmetric.

1.4.3 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 (MSE) ?? and for the binary output, we use binary cross entropy ?? . We calculate the total loss as a weighted sum of 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 the loss evenly for the remaining 50% of the weight. During the introductory phase of the training, 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 stuck to the values defined above.

1.4.4 Hypertuning

For the hypertuning we focus on architecture complexity, learning rate, momentum and weight decay. We will use the ADAM optimizer with the initial default values of $\beta_1 = 0.9$, $\beta_2 = 0.999$ and zero weight decay (we will change momentum β_1 and weight decay). We use a batch size of 32 and train the model for a maximum of 100 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 major 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 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 just a single epoch where a batch size of 32 yields a total of 242 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. 1.8 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 .

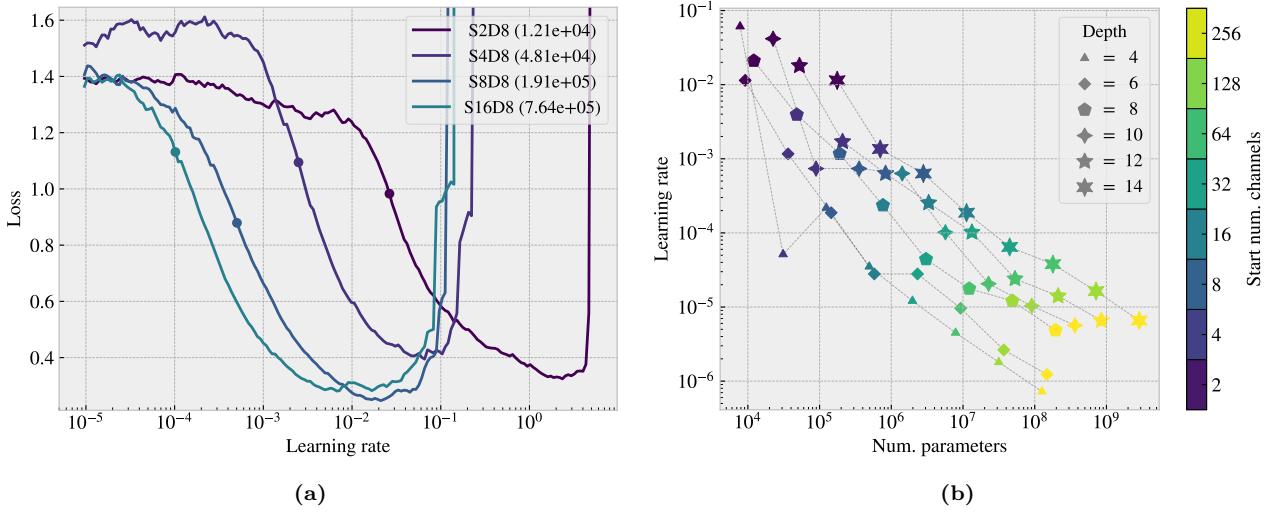


Figure 1.8: Learning rate range test for various model complexities. We increase the learning rate exponentially from 10^{-7} to 10 during one epoch corresponding to an exponent increment of roughly $1/30$ per batch iteration. (a) shows a few examples of the training loss history as a function of the learning rate. The exemplary architectures are $S[2, 16]D8$ with the corresponding number of model parameters shown in parentheses in the legend. The dot indicates the suggested learning rate at the steepest decline of the slope. (b) shows the full results of suggested learning rates depending on the number of model parameters with color coding differentiating the number of start channels and marker types differentiating different model depths.

With the use of the suggested learning rates from Fig. 1.8 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. 1.9 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 strain curves which we immediately found to be the more difficult trend to capture. The selected evaluation is shown in Fig. 1.10. Note that these configurations already are 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. 1.9.

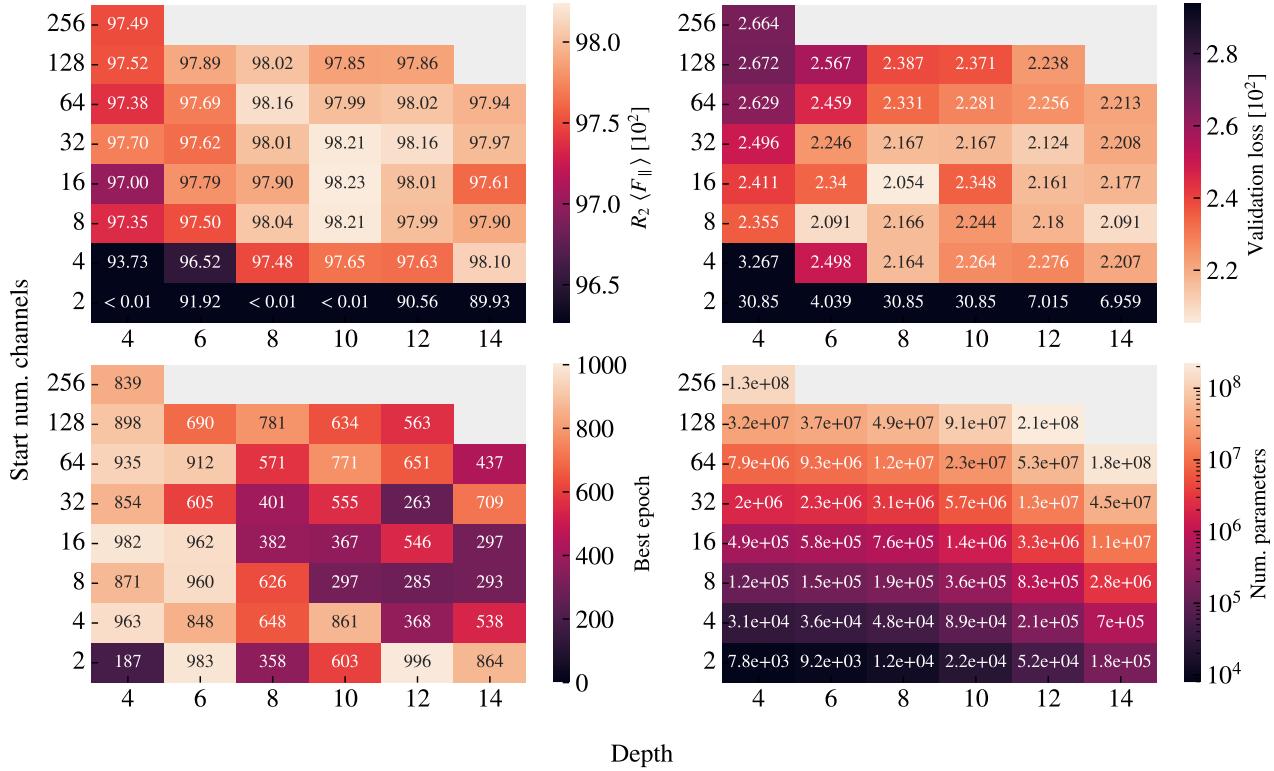


Figure 1.9: Architecture search.

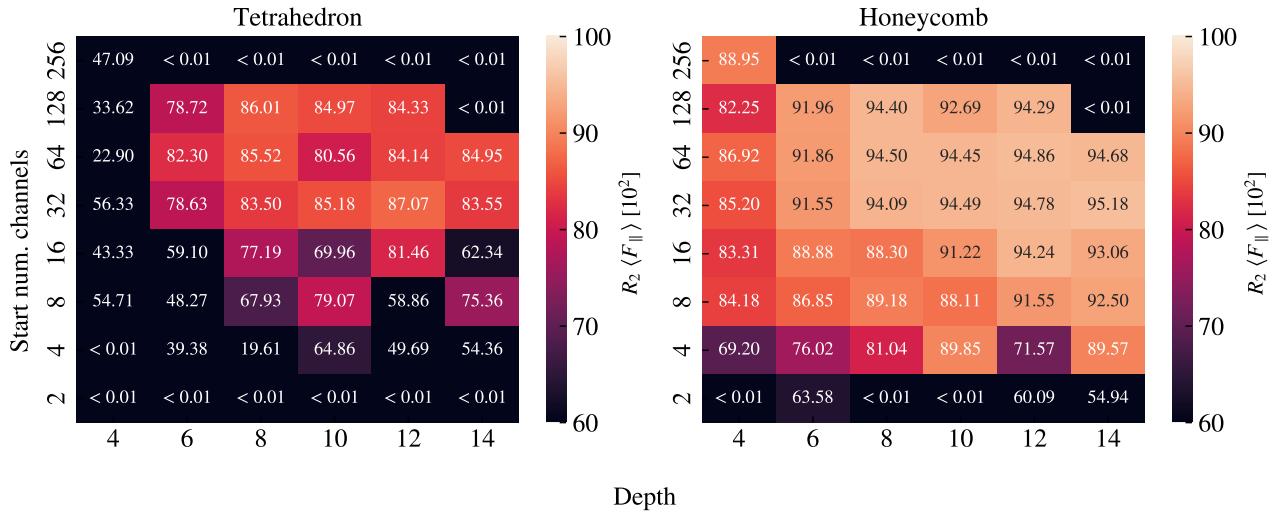


Figure 1.10: Selected pseudo validation set. Fix the missing grey fields in the top which are replaced by $\downarrow 0.01$.

From the validation scores in Fig. 1.9, 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 looking at the best epoch we find that models of low depth result in a later best epoch which is compatible with underfitting. As the depth is increased we find more models with a lower best epoch, in the range $\sim [300, 600]$, which on the other hand suggest cases of overfitting. Since our training stores the best model during training, we do not have to worry too much about overfitting, but we can take this transition from underfitting to overfitting as a sign that our search is conducted in an appropriate complexity range. When consulting the evaluation on the selected set in Fig. 1.10 we find significantly lower R_2 scores, especially for the Tetrahedron pattern. Considering, that some of these data points are also present in the training data, this shows clearly that these configurations are more challenging to predict. While the peak R_2 value for the validation score in Fig. 1.9 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 as it is among the top candidates for the R_2 score and loss in Fig. 1.9 and the R_2 score for the Honeycomb pattern in Fig. 1.10 as well. Hence, we settle on this architecture.

Next, we consider momentum m and weight decay λ in the range $m \in [0.85, 0.99]$ and $\lambda \in [0, 1 \times 10^{-2}]$. For each choice of momentum, we perform a learning rate range test. We propose two learning rate schemes: A constant learning rate as used until this point and a one-cycle policy. In the one-cycle policy, we set a maximum bound for the learning rate and start from a factor 1/20 of this bound and increase towards the maximum bound during the first 30% of the training. For the final 70% of training, we decrease towards a final minimum given as a factor $1e - 4$ of the maximum bound. The increase and decrease are done by a cosine function. The suggested learning rate for the constant learning rate scheme is once again determined by the steepest slope on the loss curve while the maximum bound used for the one-cycle policy 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. The learning rate range test for momentum is shown in Fig. 1.11. 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. 1.12 and for the cyclic scheme in Fig. 1.13.

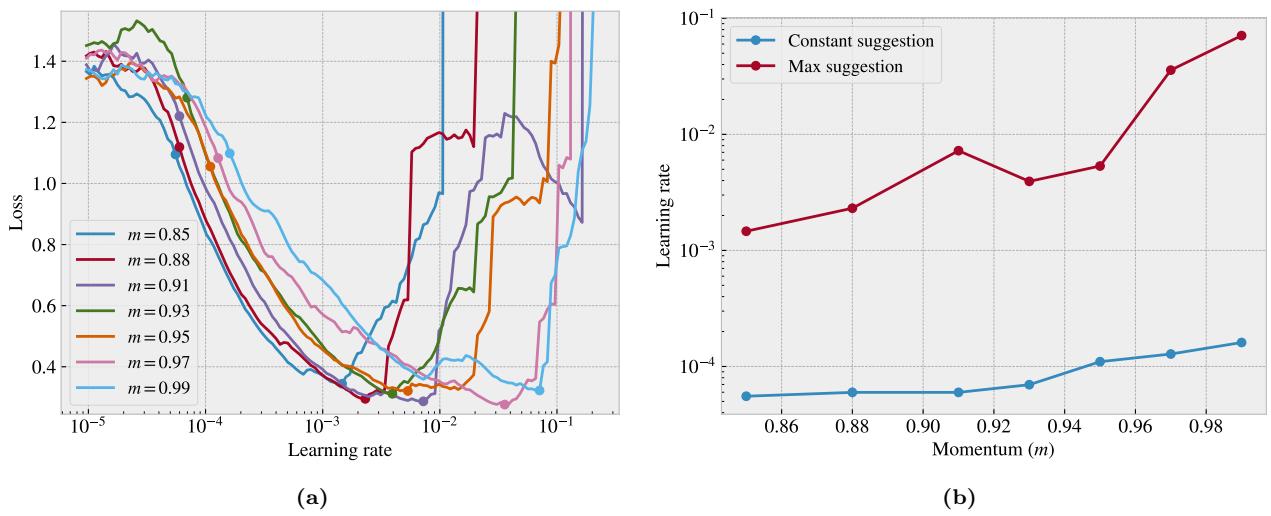


Figure 1.11: Momentum learning rate range tests

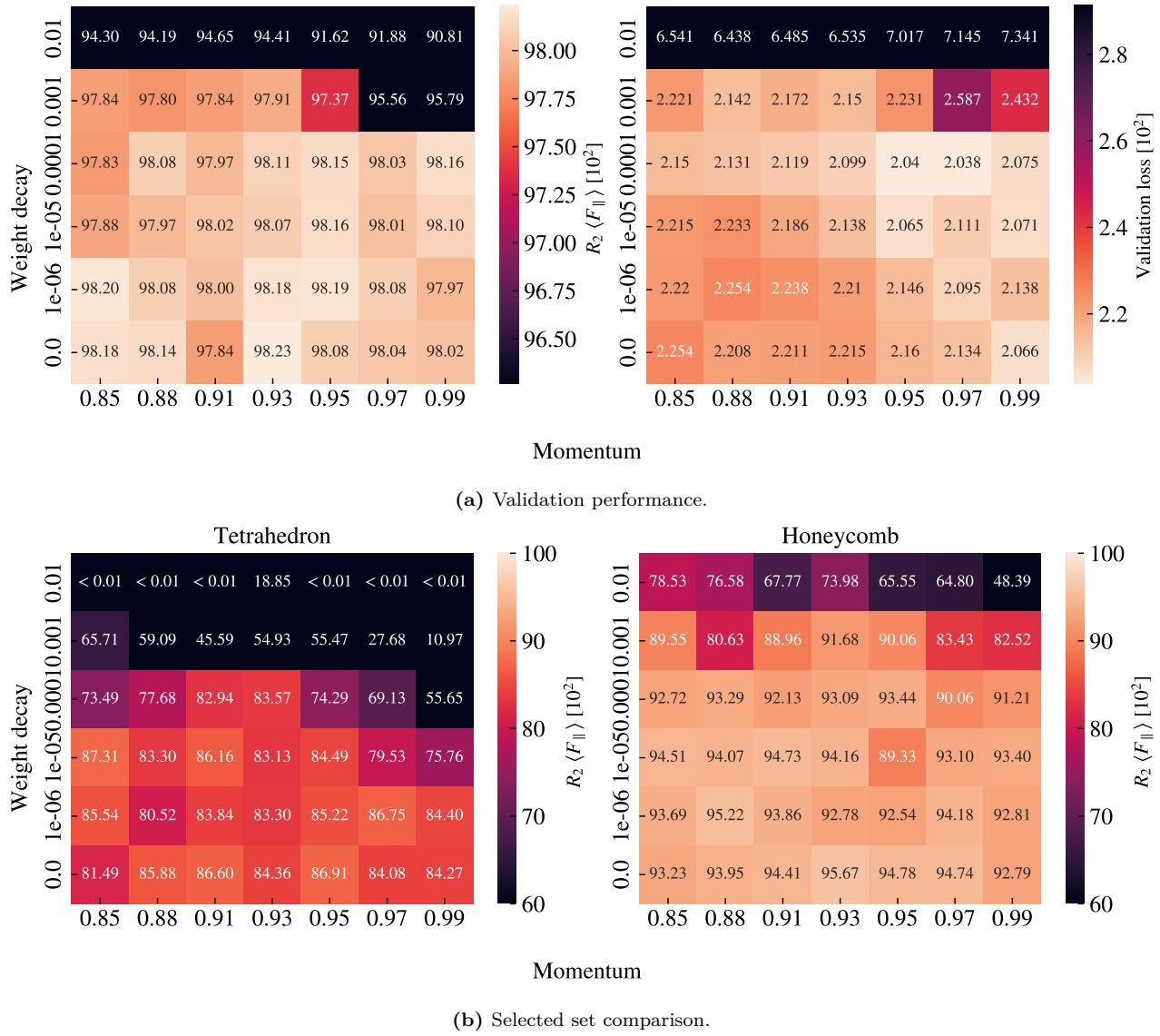


Figure 1.12: Constant learning rate and momentum scheme

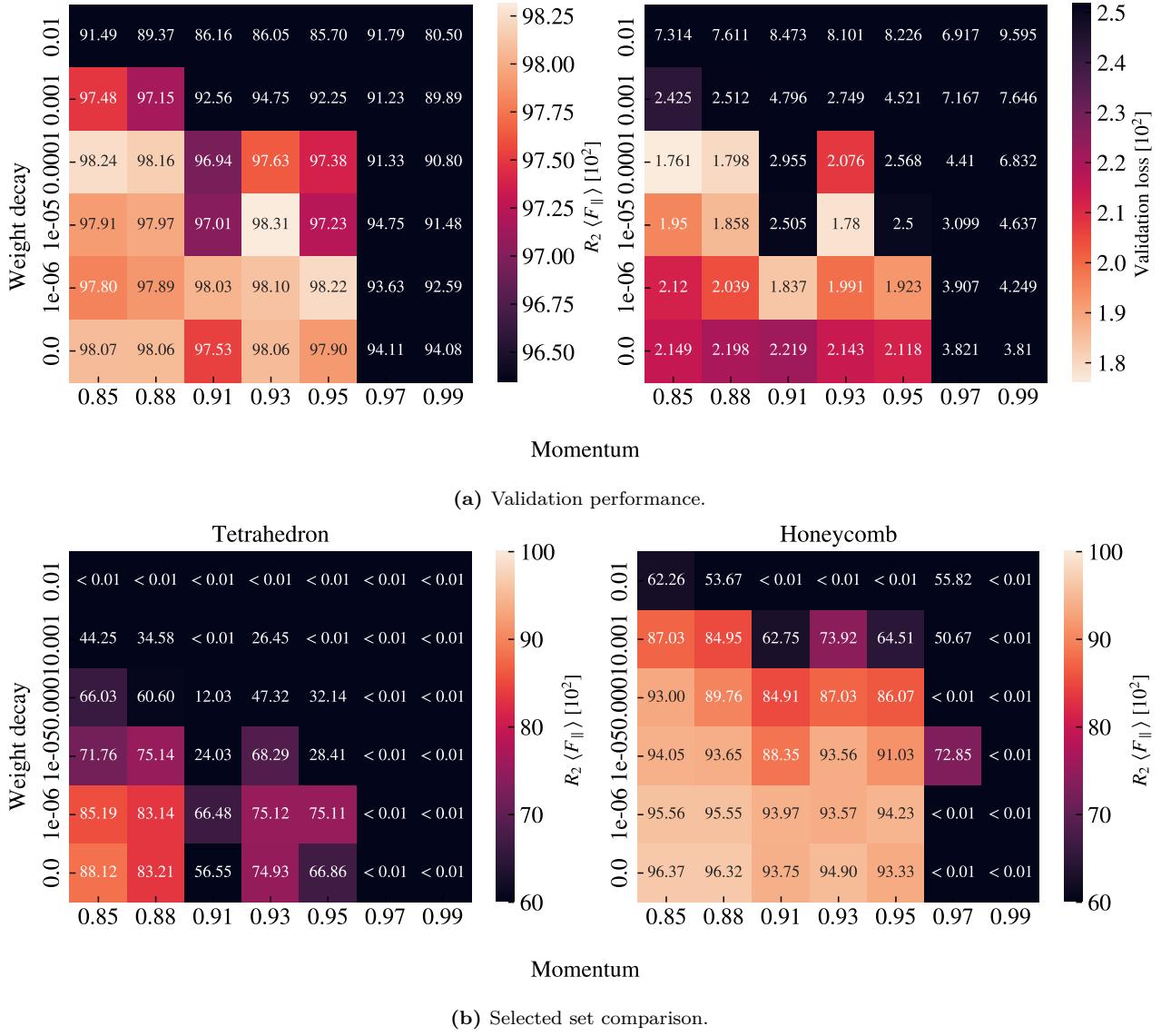


Figure 1.13: Cyclic learning rate and momentum scheme

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 can improve these scores 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). However, these values are not taken from the same hyperparameter settings and we did not find a better option across all evaluation metrics. The comparison among best scores is summarized in Table 1.3. 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 towards a low momentum $m \leq 0.93$ and low weight decay $\lambda \leq 10^{-4}$. Looking at the summary in Table 1.3 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 provided 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 absent of weight decay allow for more overfitting in general. However, for the purpose of performing an accelerated search, we find it more important to increase the chances of discovering novel designs than to reduce the chance 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 as a final score. Thus we choose the cyclic trained model with low 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, and it would have been natural to perform an extended search in that range. This was omitted due to time prioritization and the belief that the potential gain of doing so was not huge.

Table 1.3: Momentum and weight decay grid search using S32D12 model.

| | | 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 |

1.4.5 Final model

From the hypertuning process, we choose the S32D12 model trained by a cyclic training scheme with a momentum of 0.85, an accordingly chosen learning rate from the learning range test and zero weight decay. The model contains 1.3×10^7 model parameters. The main performance metrics are shown in Table 1.4 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 the accuracy score of 88% corresponding to never predicting a rupture. Looking at the relative error for the rupture strain we find a large error of $\sim 13\%$. This is lower for the Tetrahedron (5.9%) and Honeycomb (1.5%) set. Thus we might get the idea that the relative error is especially high on the validation set due to a few cases of very low rupture strains in the Random walk patterns which would shift up the average.

I find it a bit disturbing to use the “~” too much, but I would like to indicate that a round the numbers... can I omit it?

Fig. 1.14 shows the mean friction, max friction and relative contact in comparison to the Tetrahedron (7, 5, 1) and Honeycomb (2, 2, 1, 5) used in the pilot study. We note that these configurations are also partly contained within the training data, but this serves as a visualization of how the fit looks for R^2 scores above 98%. We will evaluate a true test set later on based on the proposals from the accelerated search.

Table 1.4: Mean values are used over different configurations.

| | 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 |

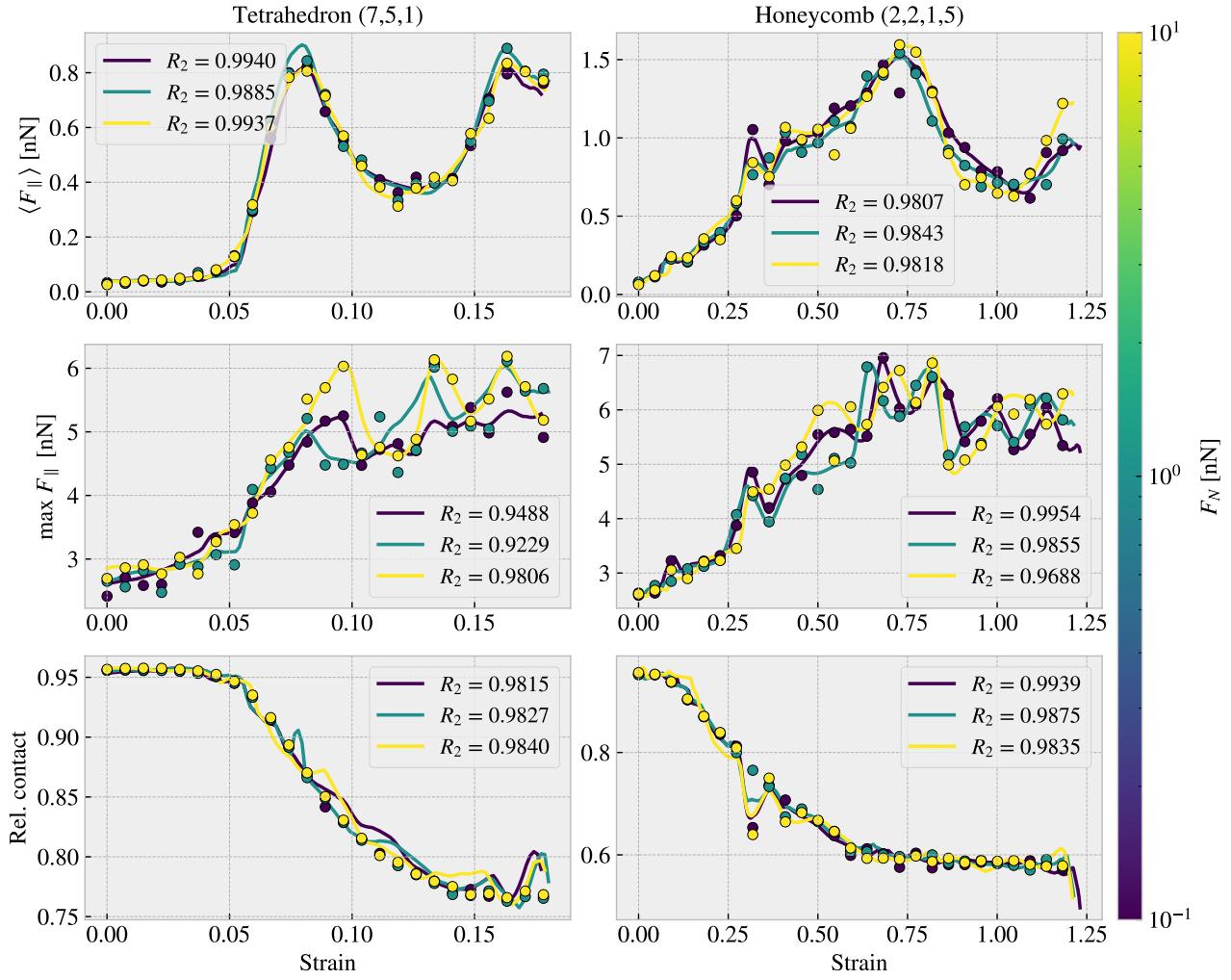


Figure 1.14: With 10^3 points in the strain range $[0, 1.5]$ and stopping after first rupture prediction.

Using 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 ??–1.7 in comparison to the actual ranking in the dataset. Generally, we find that the ML performs rather well in the ranking of the maximum properties by 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 greater than for the remaining properties. This is supported by the fact that the model predicts the Random walk 12 (Table 1.7) to have negative mean friction, showing a lack of precision. 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 in the top 5. This gives an incentive that our model can be used to perform an accelerated search of new configurations yielding a meaningful ranking of property scores.

Table 1.5: Tetrahedon

| ML Rank | Data | | ML | | Data Rank |
|------------------------------|-----------|------------|-----------|------------|--------------|
| | Config | Value [nN] | Config | Value [nN] | |
| min F_{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_{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 ΔF_{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 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 |

Table 1.6: Honeycomb

| ML Rank | Data | | ML | | Data Rank |
|------------------------------|--------------|------------|--------------|------------|--------------|
| | Config | Value [nN] | Config | Value [nN] | |
| min F_{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_{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 ΔF_{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 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 |

Table 1.7: RW

| ML Rank | Data | | ML | | Data Rank |
|------------------------------|--------|------------|--------|------------|--------------|
| | Config | Value [nN] | Config | Value [nN] | |
| min F_{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_{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 ΔF_{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 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 |

1.5 Accelerated Search

From Sec. 1.4 we have found promising results that we can use a machine learning model to predict the friction behavior. 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 optimization metrics. 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 the genetic algorithm method we perturb (mutate) the configurations and optimize for the maximum drop property using the ML model to evaluate the fitness function.

1.5.1 Patteren generation search

We utilize the pattern generators developed in ?? to create an extended dataset for our search. For the Tetrahedron and Honeycomb patterns, the increment of the parameters will lead to an increased spacing within the pattern. This 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 max 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 10 times to explore translational effects. This gives in total 135k configurations for the Tetrahedron pattern and 2025k for the Honeycomb pattern. For the Random walk generator, we do a Monte Carlo sampling. 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 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: True* and *Centering: False* throughout the search. For the handling of clustering, we implement the repair algorithm such that the sheet is repaired by the least modifications approach rather than retrying the generation several times *Make sure that this is introduced somewhere*. Due to the extra computation time associated with the random walk and the repair algorithm, we only generate 10k configurations within this class. For the ML evaluation of the configurations we use a normal load of 5 nN and generate a strain curve in the domain [0, 2] using 100 evenly spaced points. We compute the properties of interest and rank the configurations accordingly. The top candidates for each property are shown in Table 1.8 including a comparison to the original dataset top candidates from Table 1.2. The random walk top five candidates are visualized in Fig. 1.15.

Table 1.8: Pattern search. The values are 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. | 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 highlight the weaknesses of the model as we reward an unphysical negative value. In order to resolve this problem one might need 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. 1.15, while for the Tetrahedron and Honeycomb patterns, this is evident from the configuration parameters shown in Table 1.8 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.

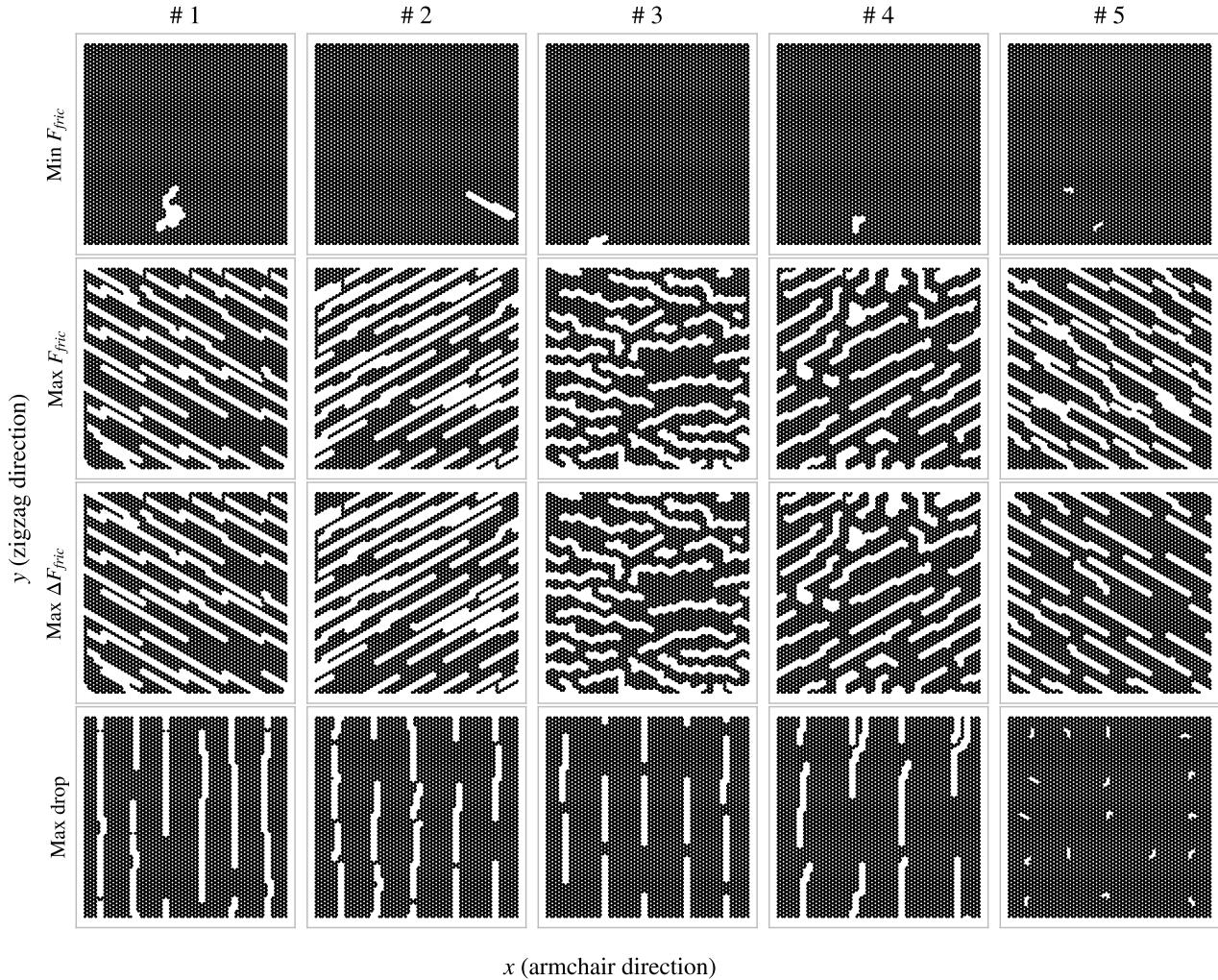


Figure 1.15: RW search top results.

Among the remaining maximum properties, we find competing values for the Honeycomb and Random walk classes. However, the top-scoring values for the Honeycomb correspond to configurations within the original dataset which is also the case for the Tetrahedron top candidates. The only difference is the randomized reference position. When taking a closer look at the 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 each pattern parameter 10 times 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 scores for a systematic change of the reference position. We generally find the max drop scores 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. 1.16. 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 ML. Even though the patterns are periodic in the x-y-plane, with a period according to the unique number of translations shown in Fig. 1.16, 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 [refer to theory or directly to the sources?](#), and thus for a sheet where the atoms on the $\pm x$ free edge 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 generate additional translational variants of the same configurations to investigate for any physical edge dependencies or otherwise strengthen the model by this data. We earmark this suggestion for another study. When considering some of the friction-strain curves we also find that the prediction of the rupture point plays an important role for the max drop property. As 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.

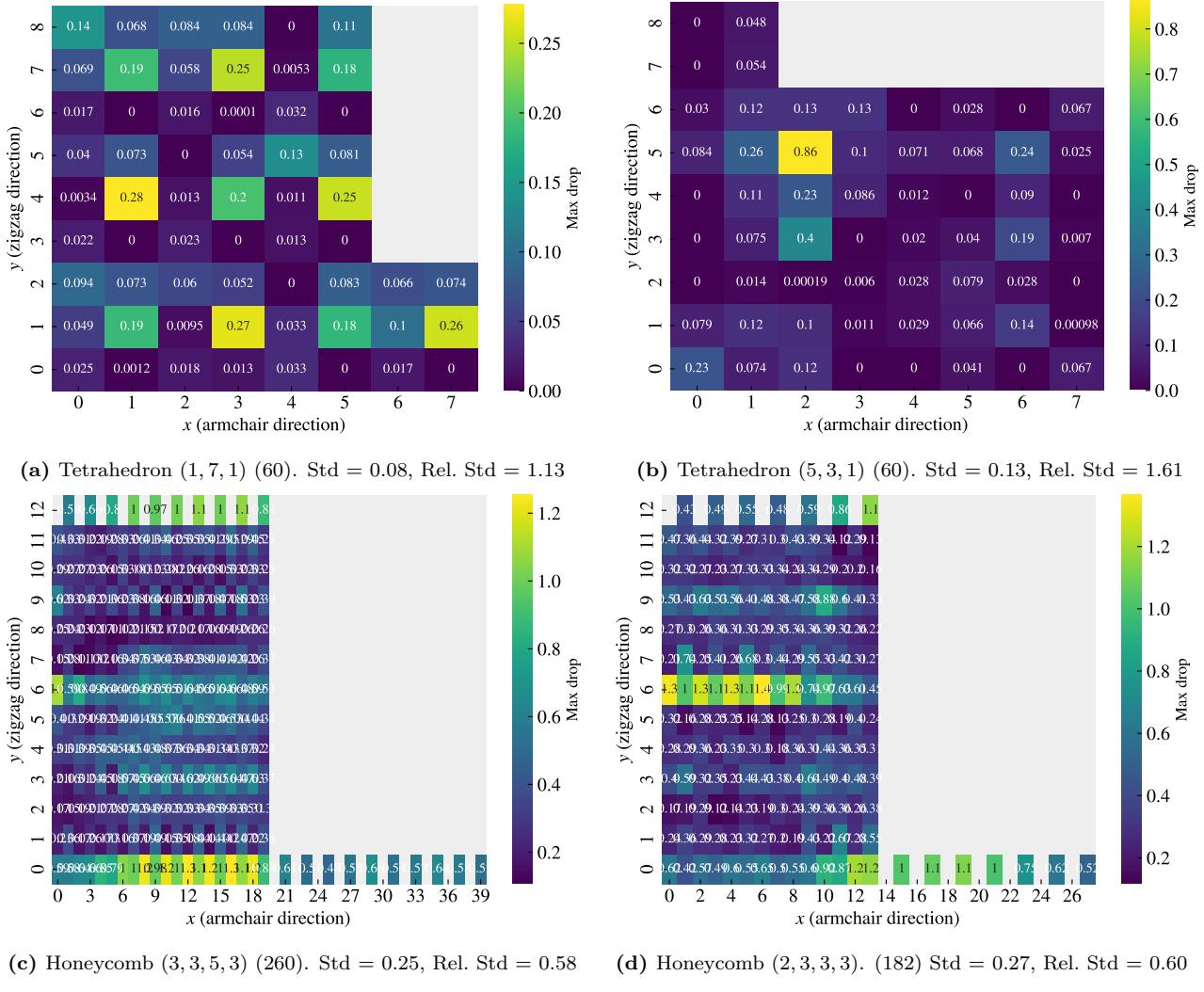


Figure 1.16: CAPTION Fix overlapping annotation text

In order to get 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 candidates for each of the properties of interest in the Random walk search shown in Fig. 1.15. We calculate the ground truth using MD simulation by sampling 30 pseudo uniform strain values using a normal load of 5 nN. Unfortunately, the test set reveals a significantly 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 scores, an average absolute error for the mean friction of 0.14 nN and a rupture accuracy of 70 %. The mean friction average R^2 score is negative which indicates that our model performs worse than simply guessing on a constant value given by 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 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. However, by going back to the hypertuning process and choosing the best model (C16D8) from the architecture complexity grid test given by

the lowest validation score, trained with a constant learning rate, we get similar results. This indicates that the poor test result is not simply caused by the hypertuning choices. Instead, it points to the fact that our original training data does not contain a generalized distribution that accurately captures 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.

1.5.2 Genetic algorithm search

Despite the fact that our machine learning analysis indicates that the model is not generalized enough for an accurate prediction on new configurations, we investigate the prospects of using 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 creating a negative friction coefficient. We verify our implementation of the genetic algorithm by optimizing a smaller 10×10 lattice for the maximum energy of the Ising model without any external field. That is, we consider the Hamiltonian

$$H = -J \sum_{\langle kl \rangle} s_k s_l,$$

for the square lattice sites s with the sum running over nearest neighbouring sites k and l . The sites take a binary value, either -1 or $+1$, and thus the highest energy for the system is reached for a checkboard pattern of every other side is -1 and the remaining being $+1$. We find that the algorithm converges toward the optimal solution within a hundred generations most of the time, which shows that it can handle some spatial dependencies in the matrix.

For the genetic algorithm search we use the top candidates from the pattern generation search as a starting point for the populations. That is, we generate the population based on the parameter associated with the max drop top candidates for the Tetrahedron, Honeycomb and Random walk search respectively. We generate a population of 100 configurations and run the search for 50 generations as we did not see much improvement for longer runs. The Tetrahedron and Honeycomb search did immediately give any useful results as the highest scoring individual from the population was not improved throughout the search even though the average score was rising initially. For the Random walk performed 5 runs based on the top 5 candidates. Most of them gave a similar result as seen for the Tetrahedron and Honeycomb patterns with only a single run providing a new generated candidate. The score of this candidate 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 which shows that we have better hopes of optimizing this property by simply generating more configurations from these parameters.

The fact that 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 porosity of 0.25 and 0.5 respectively. This time the algorithm improved the top candidate throughout, but the final top scores are still not impressive. The mixed porosity start gave the highest score, being 0.299 nN . When considering the corresponding patterns of the top five candidates in this search, they were all visually quite 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 generate any higher-level pattern structures worth further investigating. This is of course likely to be connected with the fact that the machine learning model governing the fitness evaluation is unstable outside original data domain.

By the use of the Grad-CAM method, we examine some of the proposed patterns for any noticeable trends in the model predictions. The result for the mixed porosity search top candidate is shown in Fig. 1.17. 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, shown in Fig. 1.18 to 1.19. For the mixed porosity top candidate, the Grad-CAM method highlights some areas in the noise configuration as contributing more positively than others, but we do not find any obvious trends from this. For the more structured patterns of the Tetrahedron, Honeycomb and Random walk configurations we see in many cases throughout the straining that cuts in the sheet are highlighted. This gives some confidence to the idea that the

model does consider some of the relevant features in the pattern, but it still varies too much to make any strong conclusion. However, we notice that for certain strain values, the Grad-CAM reveals considerable “attention” towards the edge of the configuration. This especially relates to the top and bottom edge ($\pm y$). For instance, Fig. 1.19 shows a highlighting of the bottom edge at strain 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 associated with the thermostat in the pull blocks might be of importance. Even though, that these results should be taken carefully due to the instability of the model, we note this as a topic for further investigations.

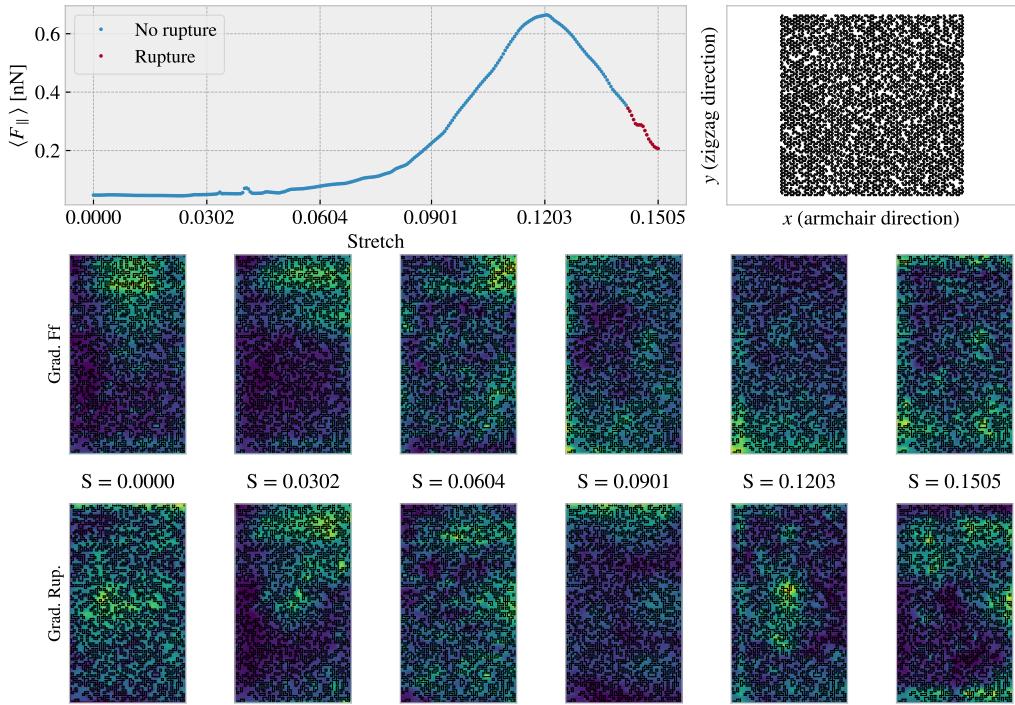


Figure 1.17: $p \in \{0.01, 0.05, 0.1, 0.2, 0.3\}$.

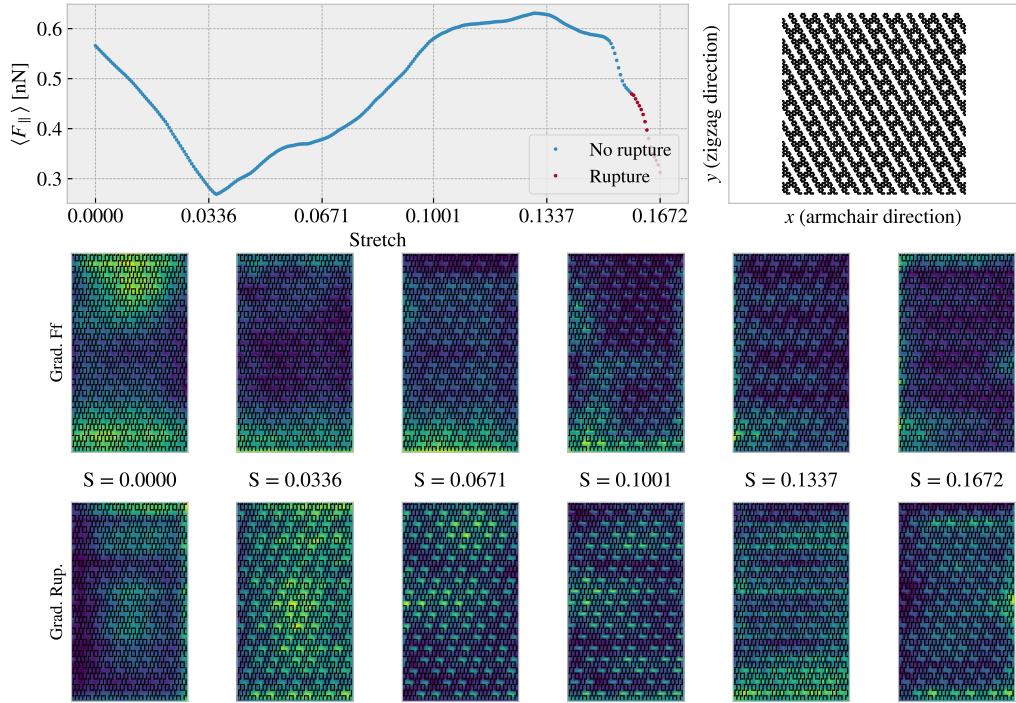


Figure 1.18: Tetrahedron (1, 7, 1), ref = (1, 4)

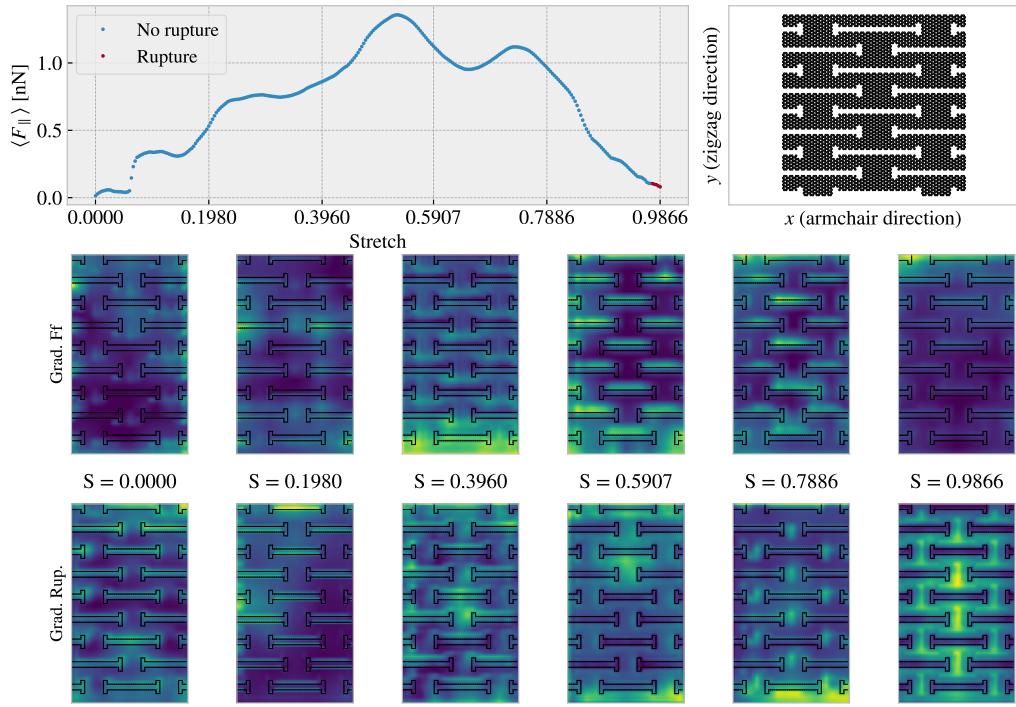
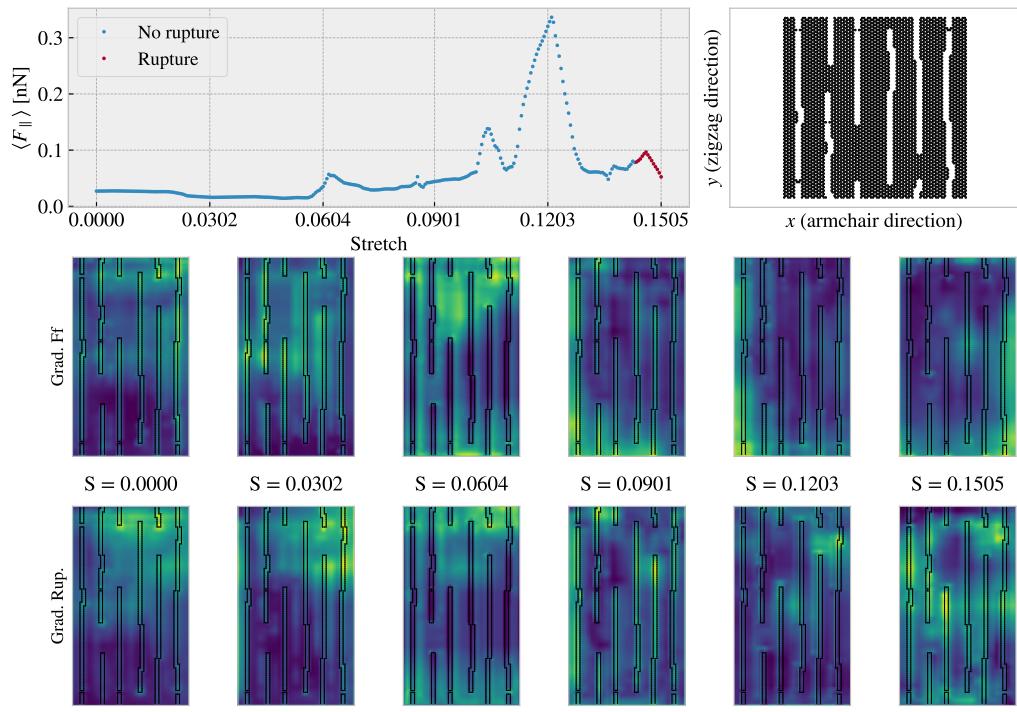


Figure 1.19: Honeycomb (3, 3, 5, 3), ref = (12, 0)

**Figure 1.20:** RW.

Chapter 2

Negative friction coefficient

For the final part of this thesis, we concern ourselves with a proof of concept approach for the designing of a negative friction coefficient. From the pilot study (??) we found the two investigated patterns to have a non-linear relationship with sheet strain. By proposing a nanomachine coupling between normal load and strain we investigate the prospects of achieving a negative friction coefficient for these patterns.

2.1 Nanomachine coupling

We do not attempt to simulate the dynamics of any nanomachine designs, but we propose that a coupling could be achieved, for instance by following a design as sketched in Fig. 2.1. This could perhaps be achieved by rigging carbon nanotubes in a similar configuration.

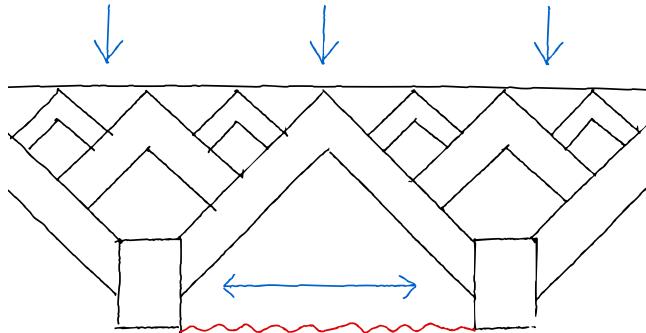


Figure 2.1: Working sketch for nanomachine

We mimic the nanomachine coupling by implementing a load-depending tension force to our MD simulations. So far, we have kept the pull block spaced by a fixed distance throughout the simulations, but now we let the pull blocks move relative to each other under the influence of a nanomachine tension force between the blocks. The tension force F_t is modeled to be proportional to the normal load $F_t = RF_N$ by a factor R which represents the ratio for the load to strain coupling. We find that a ratio of $R = 6$ will provide the necessary tension for achieving a full strain range (till rupture) within the loading range used so far in our study. We use the Tetrahedron (7, 5, 1) and Honeycomb (2, 2, 1, 5) from the pilot study and perform multiple simulations for different loads. We sample 100 pseudo uniform normal force values in the $[0, 15]$ nN which corresponds to a tension force of $[0, 90]$ nN. For the Tetrahedron pattern we increase the load by a speed of 0.015 nN/ps, but due to a rapid change in the strain for the Honeycomb pattern we reduced the loading speed to 0.0015 nN/ps and added additional points to the range of rapidly changing strain. We compare the results to that from the pilot study ?? and consider the strain-tension curve in addition as shown in Fig. 2.2.

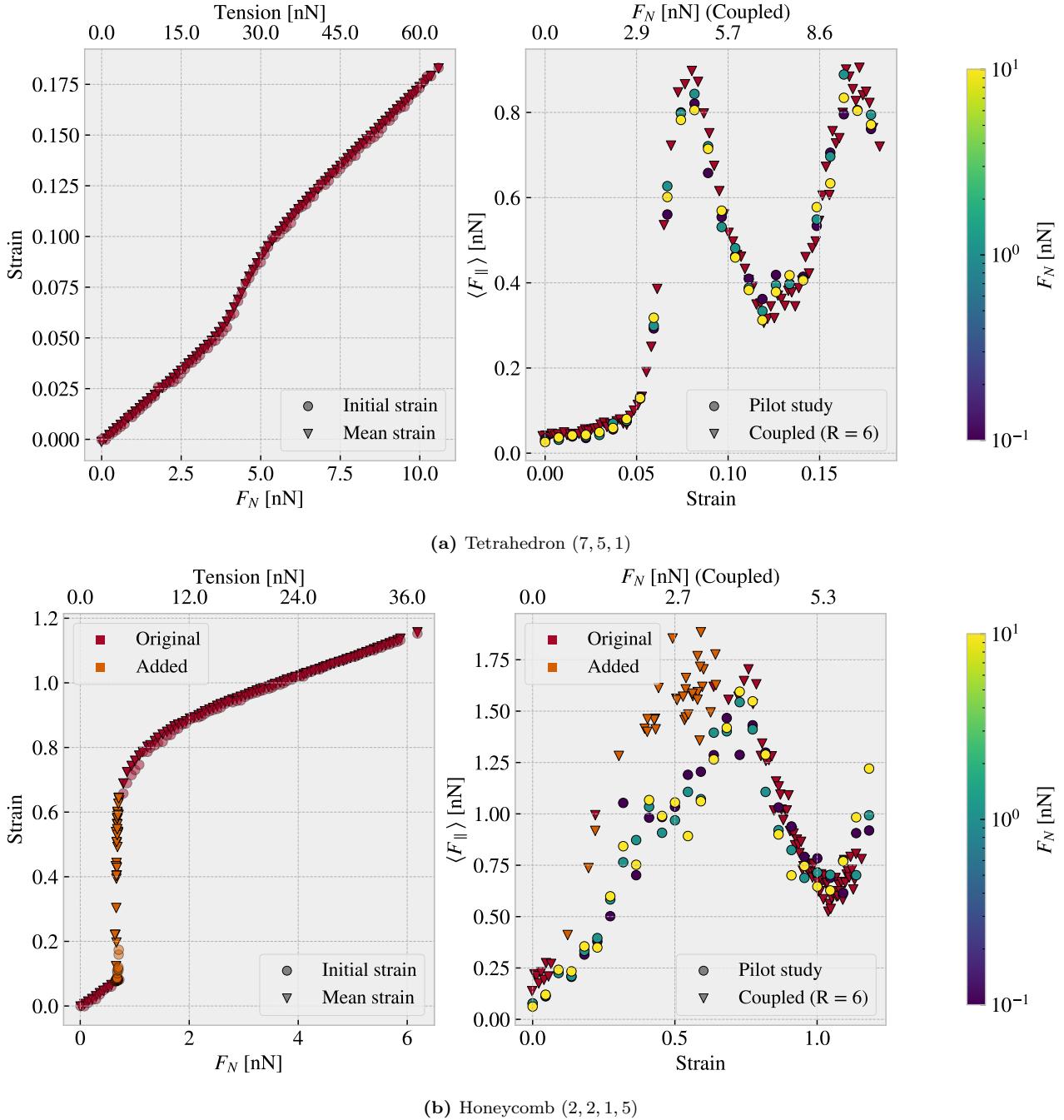


Figure 2.2: Evaluation of the friction response for a coupled system using (a) the Tetrahedron (7, 5, 1) and (b) Honeycomb (2, 2, 1, 5) pattern. The pull blocks are allowed to move relative to each other under the influence of a tension force F_t modeled as $F_t = RF_N$ for normal load F_N and a ratio $R = 6$. The right panel shows the friction vs. stretch curve in comparison to the locked pull block system and constantly defined normal from the pilot study (??). The left panel shows the strain-tension curve. For the Honeycomb pattern, we added more data points in the region where the strain-tension curve changed rapidly.

Generally, we observe from Fig. 2.2 that the coupled system follows a similar trend as seen in the pilot study. Especially the Tetrahedron pattern aligns well with the comparison data. This first of all shows that the simultaneous loading and straining of the system does not suppress the non-linear trend. In both cases, we also notice that the initial and the mean strain aligns rather well. This indicates that the sliding does not contribute to a significantly increased tension in the sheet which would otherwise lead an increased strain as

well. For the Honeycomb pattern, we find an interesting trend for the strain-tension curve. At low tension, the curve of the strain is increasing seemingly linearly with strain, however, a drastic increase in strain happens at 4.5 nN ($F_N = 0.75$ nN) transitioning from a strain of roughly 0.08 to 0.7. Eventually this settles off into a linear trend before reaching the rupture point. This reflects the reason for decreasing the loading speed and adding more data points to fill this gap in strain. When considering the friction-strain curve we notice that the coupled system results deviates most in the area of rapidly increasing strain. From a closer examination of the simulation frames in ?? we notice that the strain range covered in this rapid transition aligns rather well with the unfolding of the honeycomb pattern. More precisely, we find that the Honeycomb pattern unfolds in segments buckling one at a time after passing a minimum tension. During the unfolding phase the friction increases, but immediately starts to decrease after being fully unfolded. For further studies we might investigate such transitions in order to get a better understanding of the underlying mechanisms being in play.

Chapter 3

Summary

The work presented in this thesis covers several topics (find a better opening line?). We have created an MD simulation which enabled us to study the frictional behavior of a graphene sheet sliding on a Si substrate. In addition, we have created a numerical framework for creating Kirigami design patterns and introducing these into the friction simulations. This was used to study the effects of the out-of-plane buckling induced by a selected pair of Kirigami designs in relation to a non-cut sheet under the influence of strain. Further, we have created a dataset of various Kirigami designs for the scope of investigating the possibilities with Kirigami design. We have investigated the possibility to use machine learning on this dataset and attempted an accelerated search. Finally we look into the prospects of achieving a negative friction coefficient for a system with coupled load and stretch. In this chapter we will summarize the findings and draw some final conclusions. We will also provide some topics for further research.

3.1 Summary and conclusions

3.1.1 Design MD simulations

We have designed an MD simulation for the examination of friction for a graphene sliding on a substrate. Some of the key features for the numerical procedure were that we managed the sheet through pull blocks in the ends. We could then apply load and stretch the sheet without acting directly on the inner parts. Say something about parameter dependencies from the Pilot study.

3.1.2 Design Kirigami framework

We have designed a numerical framework for creating Kirigami designs. By defining an indexing system for the hexagonal lattice structure we were able to define the Kirigami designs as 2D matrix for numerical implementation. We digitalized two different macroscale designs, which we named the *Tetrahedron* and *Honeycomb* pattern respectively, that successfully produced out-of-plane buckling when stretched. Through a numerical framework we could create an ensemble of perturbed variations which gave approximately 135k configurations for the Tetrahedron pattern and 2025k patterns for the size of the sheet used in our study. When considering the possibility to translate the patterns this gave roughly a factor 100 more of unique perturbations. We also created a framework for creating Kirigami designs through a random walk. This was further controlled by introducing features such as bias, avoidance of existing cuts, preference to keeping a direction and procedures to repair the sheet for simulation purposes. The capabilities of the numerical framework for generating Kirigami designs was far larger than the capabilities for producing MD designs within the time constraint of this thesis. Thus we believe that this contains the possibility to benefit more extended studies and for the creation of a larger dataset.

3.1.3 Control friction using Kirigami

We have investigated the friction behavior of the non-cut sheet and a selected Tetrahedron and Honeycomb pattern under various stretch and load. The non-cut sheet did not exhibit significant out-of-plane buckling as opposed to the Tetrahedron and Honeycomb pattern. This is even when considering that the non-cut sheet had

a yield strain of 0.35 while the Tetrahedron had a lower yield strain of 0.21 and the Honeycomb a considerable larger one at 1.27 based on a stretch in vacuum. The out-of-plane buckling resulted in a significant reduction of the contact area as the sheet were stretched, towards a minimum of X for the Tetrahedron and y for the Honeycomb pattern. However, this disagreed with the asperity theory hypothesis of a decreasing friction with decreasing contact area. We found that the strain-induced buckling was initially (at low relative strain) associated with an increase in friction. Moreover, the friction-strain curve produced a non-linear behaviour which was not compatible with the approximately monotonic decreasing contact area as strain were increased. This is shown in ???. This led us to the conclusion that the contact area cannot be attributed a dominant mechanism for friction throughout the straining of the studied Kirigami sheets. In general we found a non-existing relationship between friction and load considering the uncertainties in the simulation. This is best attributed to the superlubric state of the graphene sheet on the substrate. The slope of the friction-load curves were not significantly affected by the straining of the Kirigami sheet and thus we conclude that the load effect on friction is negligible compared to the strain effects.

3.1.4 Capture trends with ML

With the use of MD simulations, we have generated an extended dataset of 9660 data points based on 216 Kirigami configurations (Tetrahedron: 68, Honeycomb: 45, Random walk: 100, Pilot study: 3) under various strains and normal loads. 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 from the pilot study suggesting that these features are relevant, but not necessarily the cause, of the observed phenomena. 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 candidates within our dataset. From these results, we found no incentive of the possibility to reduce friction with the Kirigami approach since the non-cut sheet provided the lowest overall friction. Regarding the maximum properties, we found an improvement from the original pilot study values and with the Honeycomb pattern producing the highest scores. This suggests that the data contains some relevant information for optimization with respect to these properties. 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.

For the machine learning investigation, we have implemented a VGGNet-16-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 contains 1.3×10^7 and was trained using the ADAM optimizer for a cyclic learning rate and momentum scheme for 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 Tetrahedron ($R^2 \sim 88.7\%$) and Honeycomb ($R^2 \sim 96.6\%$) pattern based on the top 10 max drop scores respectively. These scores were lower despite the fact that the selected set was partly included in the training data as well and the fact that 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 prediction. 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 constitutes the majority of the data set. By evaluating the ability for the model to rank the dataset according to the property scores we found in general a good representation of the top 3 scores for the maximum categories, while the minimum friction property ranking was lacking. We attribute this latter observation to a higher need for precision in order to rank the lowest friction values properly which the model did not possess.

In order to provide a more true evaluation of the model performance we created a test set based on MD simulations for an extended Random walk search. This test revealed a significantly worse performance than seen for the validation set with a two-order of magnitude higher loss and a negative friction mean R^2 score which corresponds to the prediction being worse than simply guessing on a constant value based on the true data mean. However, by considering one of the early hypertuning choices, regarding architecture complexity, we evaluated the model when prioritizing mainly for the lowest validation loss. This gave similar performance on the test set which indicates that it is not simply a product of a biased hypertuning process, since we based our choices on the selected configuration set (which overlapped with the training data). Instead, it points to the fact that our

original dataset did not cover a wide enough configuration distribution to accurately capture the full physical complexity of the Kirigami friction behavior.

3.1.5 Accelerated search

Using the ML 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 $135\text{ k} \times 10$ Tetrahedron, $2025\text{ k} \times 10$ Honeycomb and 10 k Random walk patterns. For the minimum friction property, the search suggests a favoring of a low cut density (low porosity) which aligns with the overall idea that the dataset does not provide an incentive for further friction reduction. The maximum properties resulted in some minor score increases but the suggested candidates were overlapping with the original dataset. By investigating the sensitivity to translation of the Tetrahedron and Honeycomb patterns we found that the model predictions varied drastically with pattern translation. This can be attributed to a physical dependency since the edge of the sheet is effected by this translation. However, due to the poor model performance on the test set, we find it more likely to be a model insufficiently arising from a lacking training dataset.

For the genetic algorithm approach, we investigated the optimization for the max drop property with respect to starting population based on the result from the extended dataset accelerated search, and some random noise initializations with different porosity values. This approach did not provide any noteworthy incentive 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. However, this is highly affected by the uncertainty of the model predictions, and thus we did not pursue this any further. By considering the Grad-CAM explanation method we found that the model predictions sometimes seem to pay considerable attention to the top and bottom edge of the configurations. This is surprising since these are not true edges but are connected to the pull blocks in the simulation. Despite the uncertainties in the predictions, we argue that this might be attributed to thermostat effects from the pull blocks and thus we note this as a feature worth more studying.

3.1.6 Negative friction coefficient

By enforcing a coupling between load and stretch, mimicking a nanomachine attached to the sheet, we investigated the load curves arising from loading of the Tetrahedron (7, 5, 1) and Honeycomb (2, 2, 1, 5) pattern from the pilot study. The non-linear trend observed for increasing strain carried over to the coupled system as well producing a highly non-linear friction-load curve. This demonstrates a negative friction coefficient say something about the values.

3.2 Outlook / Perspective

Having successfully demonstrated a non-linear effect on friction with increasing strain of the sheet our results invite a series of further studies to investigate this relation. First of all, it would be valuable to investigate how the friction-strain curve depend on temperature, sliding speed, spring constant, and on load for an increased range $F_N > 100nN$. This is especially interesting in the context up conditions leading to a stick-slip behavior as our results were carried in the smooth sliding. Moreover, it would be important to verify that the choices for relaxation time and pauses are not critical for the qualitative observations as well as trying a different interatomic potential for the graphene and perhaps an entirely different substrate material. Especially the Adaptive Intermolecular Reactive Empirical Bond Order (AIREBO) potential for the modeling of the graphene sheet might be of interest. The effects from excluding adhesion (the LJ interaction) can also be useful for the investigation of the observed phenomena.

In order to get a better understanding of the underlying mechanism for the friction-strain relationship we might investigate commensurability effects further by varying the scan angle. We might also consider investigating the friction-strain relationship under a uniform load to get insight into whether the loading distribution is of importance. Another topic worth studying is the relation to scale. Thus it would be interesting to study size effects but also further look into edge effects by translating the pattern. With this regard, we would also suggest a more detailed study of the effect from the thermostat in the pull blocks which is suggested to have a possibly importance by judging from the machine learning model Grad-CAM analysis.

For machine learning, we can either try to extend the data set to resolve the issue of the model not being generalized enough. We can also create a dataset for a single kirigami design and include some of the mentioned physical variables above and attempt to use machine learning for unraveling these relations. In that context we would advice for as more detailed investigation of machine learning techniques. If succesfull this would invite a study of inverse design methods such as GAN or diffusion models.

- How is this behavior effected by scaling?
- How does the distribution of normal load effect the Kirigami friction behavior?
- Things to vary: load range, scan directions, adhesive forces, longer relaxation time, different potential (AIREBO)
- Investigate if the contact area is effecting the friciton non-linear by turning of friction force for atoms corresopnding to those that lift off from the sheet during the out-of-plane buckling.
- Investigations of commensurability effects.
- Study dependency of translation of the patterns as suggested by the ML results.
- Investigate effects from pull blocks...
- Investigate effects from the thermostat since the top and bottom edges was shown interest by the model prediction.

Appendices

Appendix A

Appendix A

Appendix A

Appendix B

Appendix B

Appendix C

Bibliography

- ¹P. 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).
- ²P. 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).
- ³L.-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).