

Modeling On-Surface 2D Polymer Synthesis with Stochastic Simulations

McGill University Department of Physics

Zach Charlesworth, Lex Rouquette

Supervisor: Peter Grutter

April 17, 2025

Abstract

Stochastic and kinetic Monte Carlo simulations have been established in order to provide modeling tools for experimental 2D polymer formation on coinage metal substrates. The simulations take several parameters into account, such as the diffusion, rotation, coupling, and later dehalogenation energies of the monomers used to create larger structures. Additionally, configurable step and wandering defects have been implemented to increase the accuracy of the substrate surface to the experimental realizations. Furthermore, simulations were conducted using the stochastic algorithm in order to closely reproduce a polymer island which formed during an experiment, with reasonable results.

1 Introduction

Graphene is a two-dimensional lattice of carbon atoms arranged in a honeycomb formation. Its highly ordered structure gives rise to unique material properties such as an exceptional strength to weight ratio, and high electrical conductivity. However, due to it being a conductor, and having non-tunable Fermi-levels [1], graphene is unsuitable for most semiconductor applications. These pitfalls are theoretically mitigated in other two-dimensional graphene-like polymers, whose lattices are not solely built from carbon atoms. These two-dimensional graphene-like polymers are grown in the Grutter lab through a process known as Ullmann coupling, where monomers are added to a metal substrate, diffuse along its surface, and bond together to form two-dimensional polymers. When monomers are introduced to the surface, they are halogenated, meaning all bonding sites are occupied with a halogen atom. This

causes the bond to be inert until the halogen is catalytically removed by the metal substrate, forming a free radical which is stabilized by the metal substrate. Then, either carbon-carbon bonds form directly, or a metastable organometallic (OM) intermediate is formed first, before reaching the final state via thermal or photochemical activation [2].

The rate of formation, size of resulting polymers, structure of those polymers, and other growth parameters are currently subject to investigation, but the phase space of tunable parameters is vast and the time to undertake an experiment is relatively long. For this reason, construction of a computerized modeling program to accurately describe the polymer formation is of great interest in order to expedite the testing environment. Rudimentary polymer formation simulation programs exist [3, 4], but they overlook details such as substrate surface defects and extracting energy values of specific processes from comparing simulations to experimental results, such as those associated with diffusion, coupling, rotation, and dehalogenation.

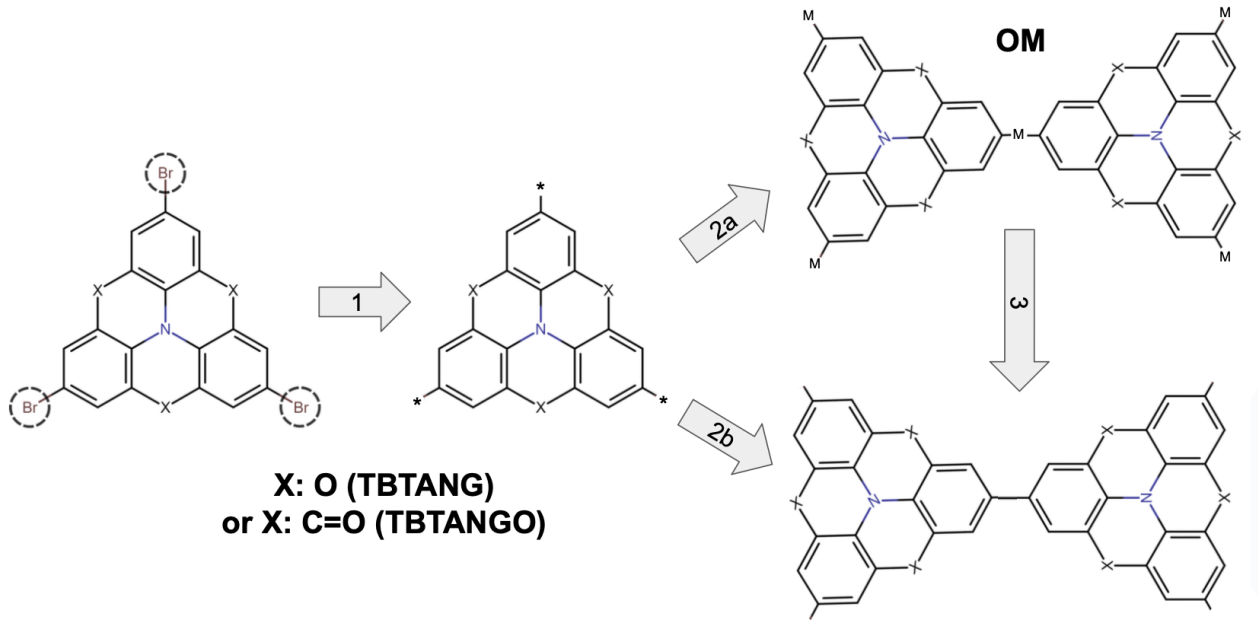


Figure 1: Schema showing the process of Ullmann coupling with the monomers used in polymer formation in the Grutter lab (TBTANG and TBTANGO). Step 1 shows the catalyzed debromination, step 2a shows the formation of the intermediate OM state, and steps 2a and 3 show the formation of the bonded monomers. M is the metal substrate which catalyzes the reaction.

1.1 Simulation Overview

We simulate the growth of a polymer from individual monomers using a stochastic simulation. The simulation is written in Python using an object oriented framework. The two main classes are `Lattice` and `Monomer`. The lattice class deals with the surface on which the monomers exist. The surface is modeled as $n \times m$ mesh grid of occupational sites with square, or hexagonal rotational symmetries, depending on the lattice structure of the substrate surfaces. As most investigations of on-surface Ullmann reactions focus on three-fold symmetric precursors on metallic (111) surfaces, we primarily focused on a hexagonal grid layout of occupation sites. Monomers can “hop” from point to point on the grid which mimics their diffusion across a metal surface. The grids coordinates are wrapped, meaning if a monomer travels off one edge of the surface it will reappear on the opposite side of the grid. The lattice class also allows step defects to be added during the grid initialization – specific details of this are outlined in section 2.3.

The `Monomer` class is responsible for the creation, diffusion, dehalogenation, and bonding of monomers. When a monomer is initialized, its energies, rates, initial orientation, and location on the lattice are specified. The monomer has three bonding sites, equally spaced around its edges. Each bonding site is halogenated when initialized, meaning it cannot couple until it has been dehalogenated.

The monomer then iterates through a series of three steps: diffusion, rotation, and coupling. Additionally, each timestep includes a dehalogenation step, in which every monomer on the lattice can undergo dehalogenation. The probability of the monomer diffusing, coupling, rotating, or dehalogenating is set by

$$P_i = R_i^{-\frac{E_i}{k_B T}} \Delta t, \tag{1}$$

where P_i is the probability of performing a certain action, R_i is the rate of that action, E_i is the action’s energy, and Δt is the simulation timestep. Here, E_i can be the diffusion energy E_D , the rotation energy E_R , the coupling energy E_C , or the dehalogenation energy E_{De} . If the monomer diffuses, it is equally likely to move in any allowed direction, as long as it not near any defects on the lattice surface. Upon rotation, the monomer is equally likely to rotate

to any allowed orientation. The coupling step involves searching for neighbouring monomers, checking their orientation, and calculating a probability of the two monomers coupling. To check if coupling is allowed, the specific bonding sites involved in the coupling reaction must be identified. This is done by calculating the relative angle of the bonding monomers, and their respective orientations. The angle is calculated using simple trigonometry:

$$\theta_r = \arctan\left(\frac{x_2 - x_1}{y_2 - y_1}\right) \quad (2)$$

Where θ_r is the relative angle, x_1 and y_1 are the coordinates of the first monomer, and x_2 and y_2 are the coordinates of the second monomer. Next, the orientation is calculated using the parameter R_{tot} .

$$R_{tot} \begin{cases} + = 1 & \text{if clockwise rotation} \\ - = 1 & \text{if counterclockwise rotation} \end{cases} \quad (3)$$

By taking $\text{mod}_6(R_{tot})$, the exact orientation can be determined. There are six possible orientations, shown in Fig. 2.

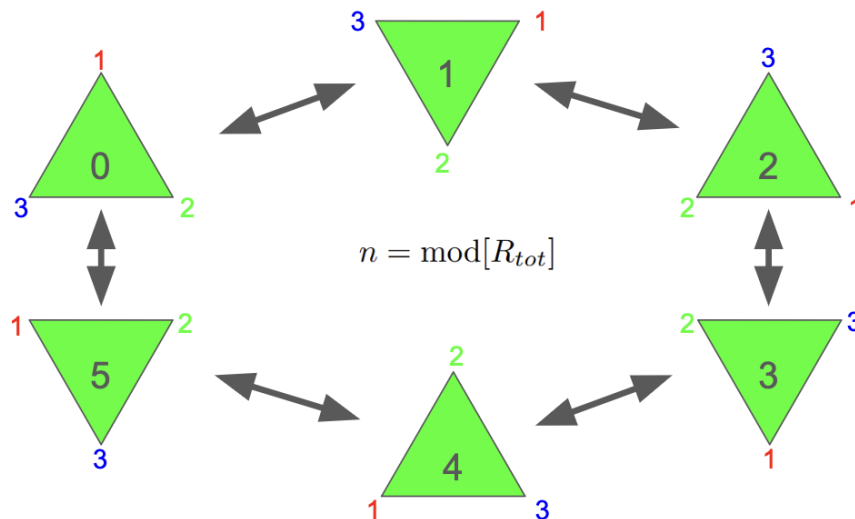


Figure 2: A schematic showing the six possible orientations of monomers in the simulation, labeled 0 through 5. The dehalogenation sites on the corners of the monomer are labeled 1 to 3, which are applied in the simulation to determine the coupling pair.

Lastly, by using the relative angle and the orientation of each monomer, the halogenation of the bonding sites used in the coupling reaction can be determined.

If a monomer couples, it will stop diffusing and rotating around the lattice. At this point, a new monomer can be introduced to the lattice, where it too will continuously diffuse and rotate until it couples to another monomer. In the event that a monomer has not coupled after a specified amount of steps (typically of the order of 10^5 to 10^7), it will be removed from the surface, allowing a new monomer to be placed on the lattice. Allowing one monomer at a time to diffuse around the lattice at a time is a reasonable approximation because the general size of the simulation substrate is on the order of 10^3 atoms, which is sufficiently small so as to assume that the flux of monomers will only permit one new monomer to act in the region.

2 Defects and Steps

Substrate surfaces are rarely flat, even over short distances. This is because of symmetry breaking at the interface between the crystalline substrate and the vacuum directly above it. This loss of crystal periodicity causes the creation of unpredictable defects – single atoms protruding or receding from the surface, known as interstitials or vacancies, respectively. These defects can act as nucleation sites for diffusing monomers, producing small islands randomly over the substrate. It is also possible for these defects to be mobile.

2.1 Wandering Defects

Addressing the existence of wandering defects in the simulation required some approximations. When a monomer nucleates at a defect, the defect is no longer allowed to move. In the time steps considered in the simulation, the probability for the defect to continue moving with a monomer coupled is very small. Additionally, monomers are not allowed to decouple from defects, nor each other.

In Fig. 3, honeycomb-like structures form around the substrate defects, similar to experimental observations. The input parameters used to generate this output are shown in Table 1.

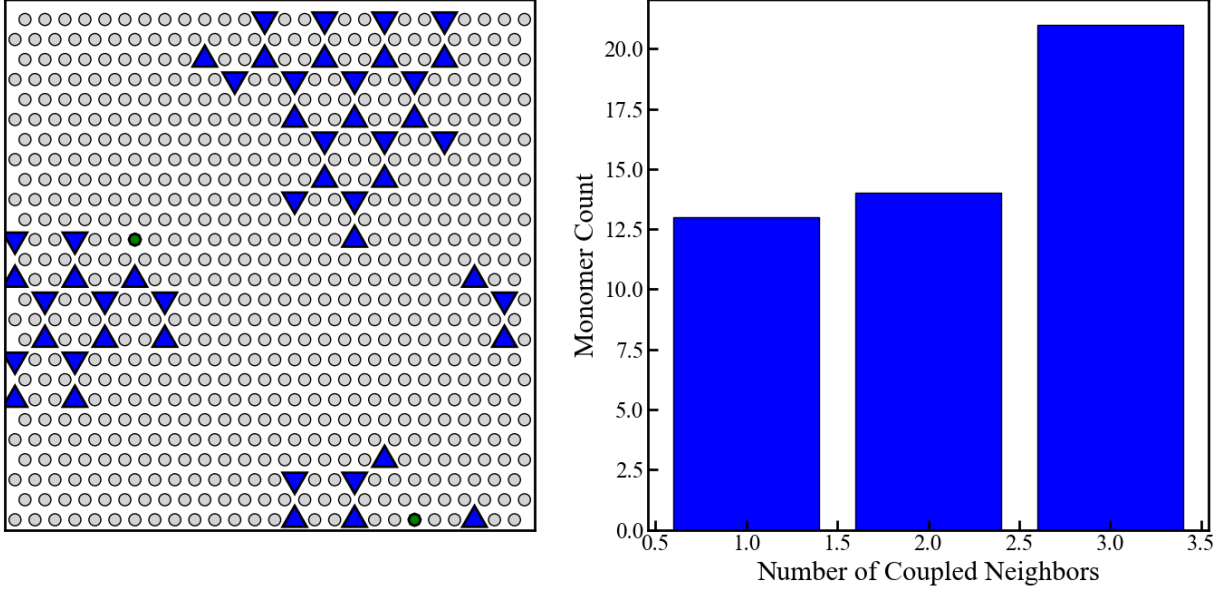


Figure 3: Left: Simulation results for 2 wandering defects (green) and 50 monomers (blue). Two distinct islands have formed, with the one on the bottom passing over the periodic boundary back to the top of the window, and a similar effect taking place with the island on the left. Green defects indicate those which have successfully coupled, while red indicate those that have not coupled. Right: Histogram counting monomers that have 1, 2, and 3 coupled neighbors.

Monomers	
Diffusion Rate	0.05
Diffusion Energy	0.0
Rotation Rate	1.0
Rotation Energy	0.1
Coupling Rate	0.5
Coupling Energy	0.0
Number of Monomers	50
Defects	
Diffusion Rate	1.0
Diffusion Energy	0.0
Defect Density	0.001
Grid Size	26

Table 1: Simulation parameters used to produce the islands shown in Fig. 3.

2.2 Herringbone Defects

When a clean gold surface is prepared in ultra-high vacuum, the top layer is not perfectly flat — instead, it rearranges itself into a zigzag pattern called the herringbone reconstruction [5]. These structures cause regular defects around the elbows, which enable nucleation similar to the wandering defects. STM image analysis carried out by Simon Briesenick found that islands form in a pattern resembling the herringbone structure of Au(111). Object classification using *ilastik* [6] revealed a mean nearest-neighbor distance of $17.0 \pm 0.3\text{nm}$, which is consistent with half the herringbone period of Au (111) – 32nm [7].

The implementation of herringbone defects was done by applying the same defect logic from Sec. 2.1, but restricting them to be immobile and periodic. The code enables specification of herringbone periodicity and separation, but the only substrate exhibiting herringbone defects used is Au (111), so the settings for the simulation may be truncated to restrict the herringbone defects for other cases. An example of this as compared to the experimental result is shown in Fig. 4.

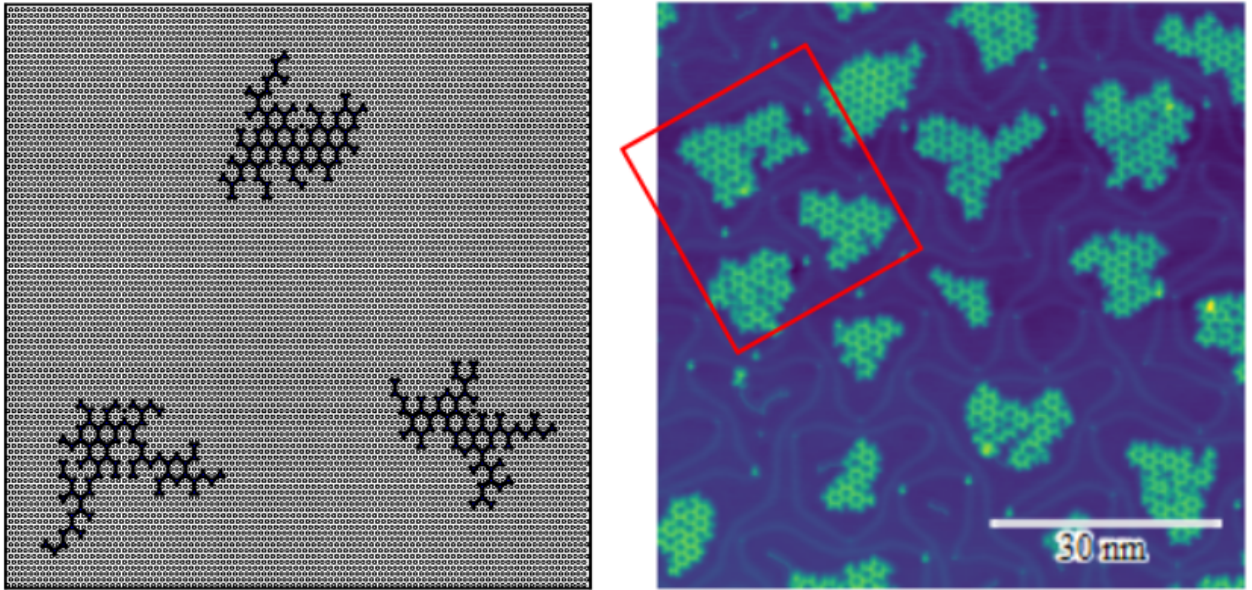


Figure 4: Example of a simulation compared to experimental results; details will be discussed in Sec. 2.2. Left: Island formation around regular herringbone elbow defects. The parameters used to create this simulation is shown in Table 2. Right: Experimental observations of island formation on Au (111). Red indicates the herringbone elbow defects that the simulation is based on.

Monomers	
Diffusion Rate	0.05
Diffusion Energy	0.0
Rotation Rate	1.0
Rotation Energy	0.1
Coupling Rate	0.5
Coupling Energy	0.0
Number of Monomers	200
Herringbone	
Start Position X	20
Start Position Y	30
Period	60
Width	50
Grid Size	100

Table 2: Parameters for the herringbone nucleation simulation shown in Fig. 4.

2.3 Step Defects

Another common type of defect on metal surfaces are step defects, which, as the name suggests, are ridges along the metal surface separating an upper elevated and lower elevated region. When a monomer is near a step, it no longer diffuses isotropically; it is more likely to travel parallel to the step, or to couple to the step, stopping the monomer from diffusing. Probabilities of diffusing over the step are skewed depending on the direction of travel of the monomer, where it is more likely for the monomer diffuse over a step when traveling from high ground to low ground, rather than vice-versa [8]. Because of the relatively high probability of monomers coupling to the step, or traveling along its ridge, it is common in experiment to see elongated structures forming along step defects.

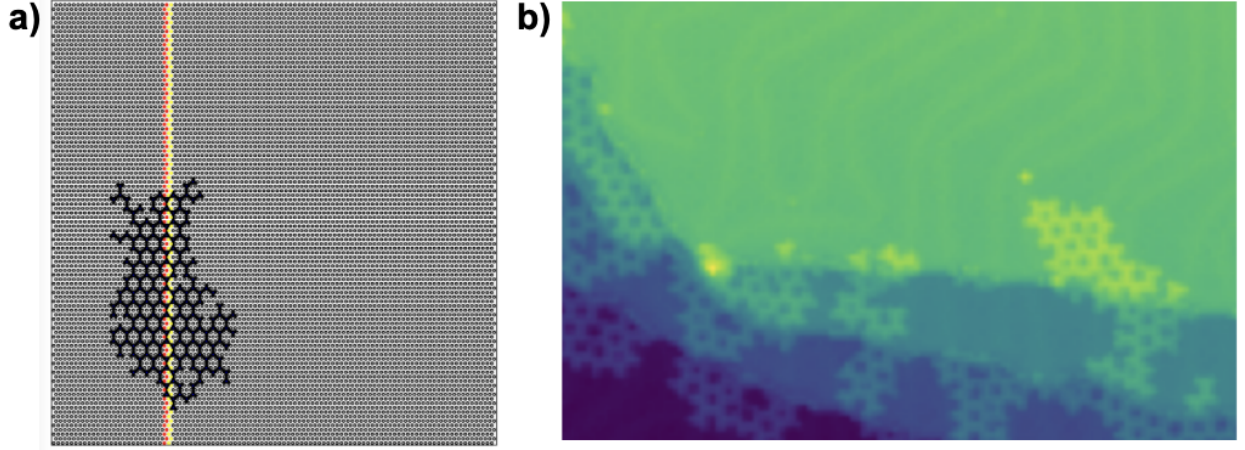


Figure 5: a) A preliminary simulation with untuned energy values modeling polymer formation along a step defect. b) An experimental observation of elongated polymer growth along step defects [7]. Since colour represents height, steps are characterized by the boundaries between different colours.

Step defects are modeled in the simulation by creating a grid of twos, ones, and zeros. The twos represent lattice sites neighbouring the upper edge of the step, the ones represent lattice sites neighbouring the lower edge of the step, and the zeros represent all other lattice sites away from the step defect. With this setup, the step is located at the intersection of the twos and ones.

2.3.1 Diffusion probability away from a step

When a monomer is on a zero lattice site, P_D sets the probability that the monomer will move. If it does move, it is equally likely to move in all directions, with probabilities set by Eq. 1.

2.3.2 Diffusion probability below a step

When a monomer is on a one lattice site, its probabilities to move in directions over the step or away from the step are damped by dampening factors:

$$P_{1S} = \frac{n_2}{n_D} p_2 P_D \quad (4)$$

$$P_{1A} = \frac{n_0}{n_D} p_0 P_D \quad (5)$$

Where P_{1S} is the probability of moving over the step from a one lattice site, P_{1A} is the probability of moving away the step from a one lattice site, n_2 is the number of directions there is a step, n_0 is the number of directions away from the step, n_D is the total number of directions a monomer can move, p_2 is a multiplier that dampens the diffusion probability in the directions towards the step, p_0 is a multiplier that damps the diffusion probability in the directions away from the step, and P_D is the diffusion probability. Probability to move along the step is then the regular diffusion probability, scaled by the number of available sites

$$P_{1X} = \frac{n_1}{n_D} P_D, \quad (6)$$

where P_{1X} is the probability of moving along the step, and n_1 is the number of directions along the step. The remaining probability is taken to be the probability of the monomer coupling to the step, which is defined as

$$P_{1C} = 1 - (P_{1S} + P_{1A} + P_{1X}) \quad (7)$$

2.3.3 Diffusion probability above a step

Similar to when the monomer is below a step, the diffusion probabilities of a monomer when it is above a step are adjusted by damping factors as follows:

$$P_{2S} = \frac{n_1}{n_D} p_1 P_D, \quad (8)$$

$$P_{2A} = \frac{n_0}{n_D} p_0 P_D, \quad (9)$$

$$P_{2X} = \frac{n_1}{n_D} P_D = P_{1X}, \quad (10)$$

$$P_{2C} = 1 - (P_{2S} + P_{2A} + P_{2X}), \quad (11)$$

where p_1 is a multiplier that damps the diffusion probability in the directions towards the step, P_{2S} is the probability of moving over the step from a two lattice site, P_{2A} is the probability

of away from a step from a two lattice site, P_{2X} is the probability of moving along the step from a two lattice site, and P_{2C} is the probability of coupling to the step from a two lattice site.

3 kMC Overview

Events in kinetic Monte Carlo simulations are selected with probabilities proportional to their respective rates. To do this, the total rate of all possible events is calculated using $R = \sum_i r_i$, where R is the total rate, and r_i is the rate of event i . An event is selected by the equation

$$\sum_{i=1}^{j-1} r_i < u_1 R \leq \sum_{i=1}^j r_i, \quad (12)$$

where u_1 is a random number between 0 and 1, and j is the index of the selected event. Event j is performed, then time is advanced by the timestep

$$\Delta t = \frac{1}{R} \ln(u_2), \quad (13)$$

where u_2 is another random number between 0 and 1. This process is repeated until the simulation is completed.

3.1 kMC Implementation

Kinetic Monte Carlo methods are implemented in the class `RateContainer`. Rates are calculated using the Arrhenius equation, scaled by the amount of options available to the monomer. Diffusion, coupling, and rotation rates are calculated by the expression

$$r_i = N_i \frac{P_i}{\Delta t}, \quad (14)$$

where r_i is the rate, P_i is the probability calculated in Eq. 1, and N_i is the number of legal actions that can be performed. Note that both diffusion and rotation rates are set to 0 once a monomer has coupled.

Kinetic Monte Carlo methods were implemented into the simulation to more accurately capture the temporal dynamics of processes occurring on vastly different timescales, such as frequent diffusion events versus rare coupling events, without relying on arbitrarily small or fixed timesteps. However, the implementation of kMC methods greatly increased the simulation time for unbalanced parameter inputs causing it to be ineffective for running simulations over large parameter spaces.

4 Energy Quantification and the Wasserstein Distance

The central purpose of this project is to produce a simulation that can be directly compared to experimental results in order to reduce the need for performing additional time-intensive experiments. Tuning of the action rates by altering their energies results in varying levels of island “branchiness” (Fig. 6), which can be quantified by analyzing the connectivity between all monomers.

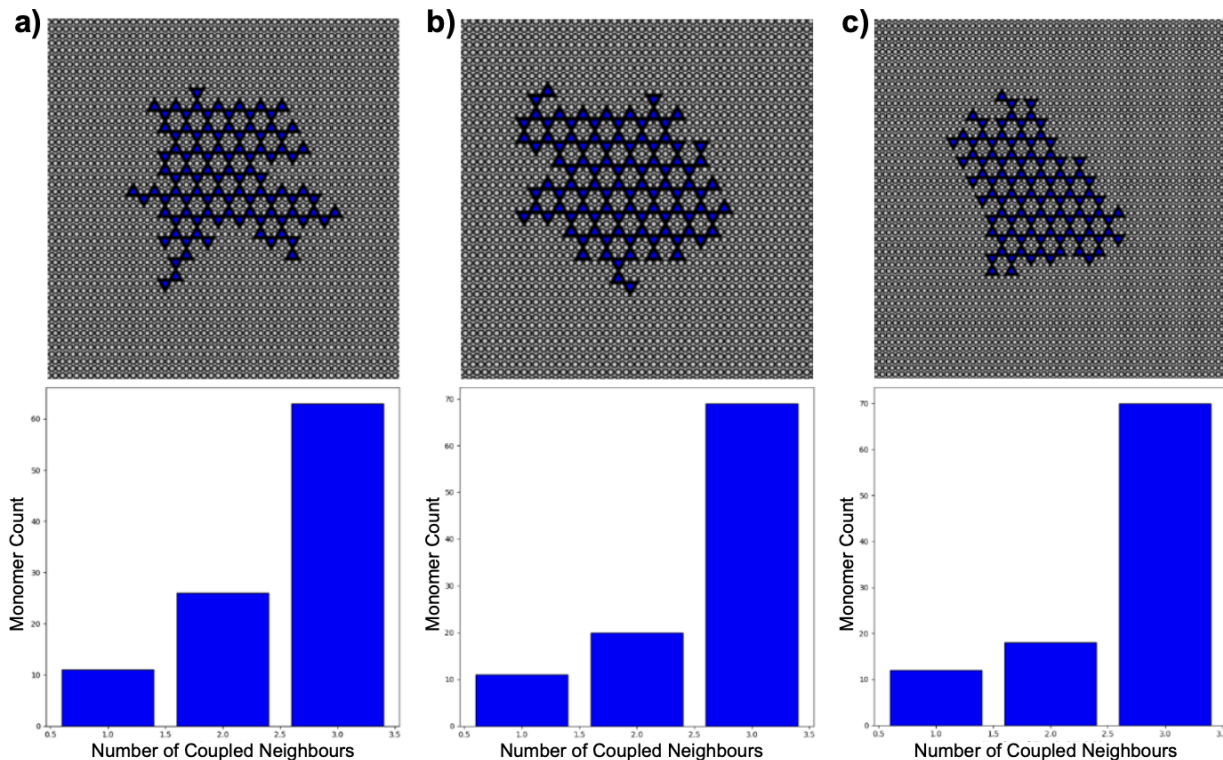


Figure 6: Single polymer structures simulated with increasing values for diffusion, rotation, coupling, and dehalogenation energies. a), b), and c) correspond to 0, 0.5, and 1.2 eV for all energies. The node connectivity histograms are shown beneath each polymer image.

The simulation program produces histograms of node connectivity, so comparing the results to an experimental scanning tunneling microscopy (STM) image requires the derivation of this node connectivity from the images.

The creation of this analysis program was undertaken by the supervising graduate student Simon Briesenick, and it extracts histograms of node connectivity by skeletonizing the intensity contrast and identifying monomers through the observation of high-connectivity nodes within the image skeleton (Fig. 7).

Histograms are not objects that can be compared to one another quantitatively using a simple difference, so a more complex model is required. Comparison of the node connectivity histograms from the simulations and experiments was done by applying the Wasserstein distance, a mathematical solution to the “soil-movers problem”. If two histograms are similar, the Wasserstein distance will reduce close to zero, whereas large differences will inform large Wasserstein distances. Over the course of the simulations, a Wasserstein distance of over 0.2 indicates large deviation, while distances reaching 0.05 were considered close matches. The p-Wasserstein distance can be written as [9]

$$W_p(P, Q) = \left(\inf_{\pi \in \Gamma(P, Q)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|X - Y\|^p d\pi \right)^{1/p}, \quad (15)$$

where $\Gamma(P, Q)$ is the set of transportation plans between the parameters making up the subspaces P and Q . For the comparison of the discrete histograms generated by the simulations and experimental STM images, the first-order discrete Wasserstein distance was used [7], written

$$W_1(P, Q) = \min_{\pi \in \Gamma(P, Q)} \sum_{i=1}^N \sum_{j=1}^N \pi_{ij} (x_i - x_j). \quad (16)$$

The `scipy` [10] implementation of the Wasserstein distance was used in the analysis of the simulations.

The optimization of the system energies was done by conducting multiple simulations over a three dimensional parameter space, illustrated in Figure 8.

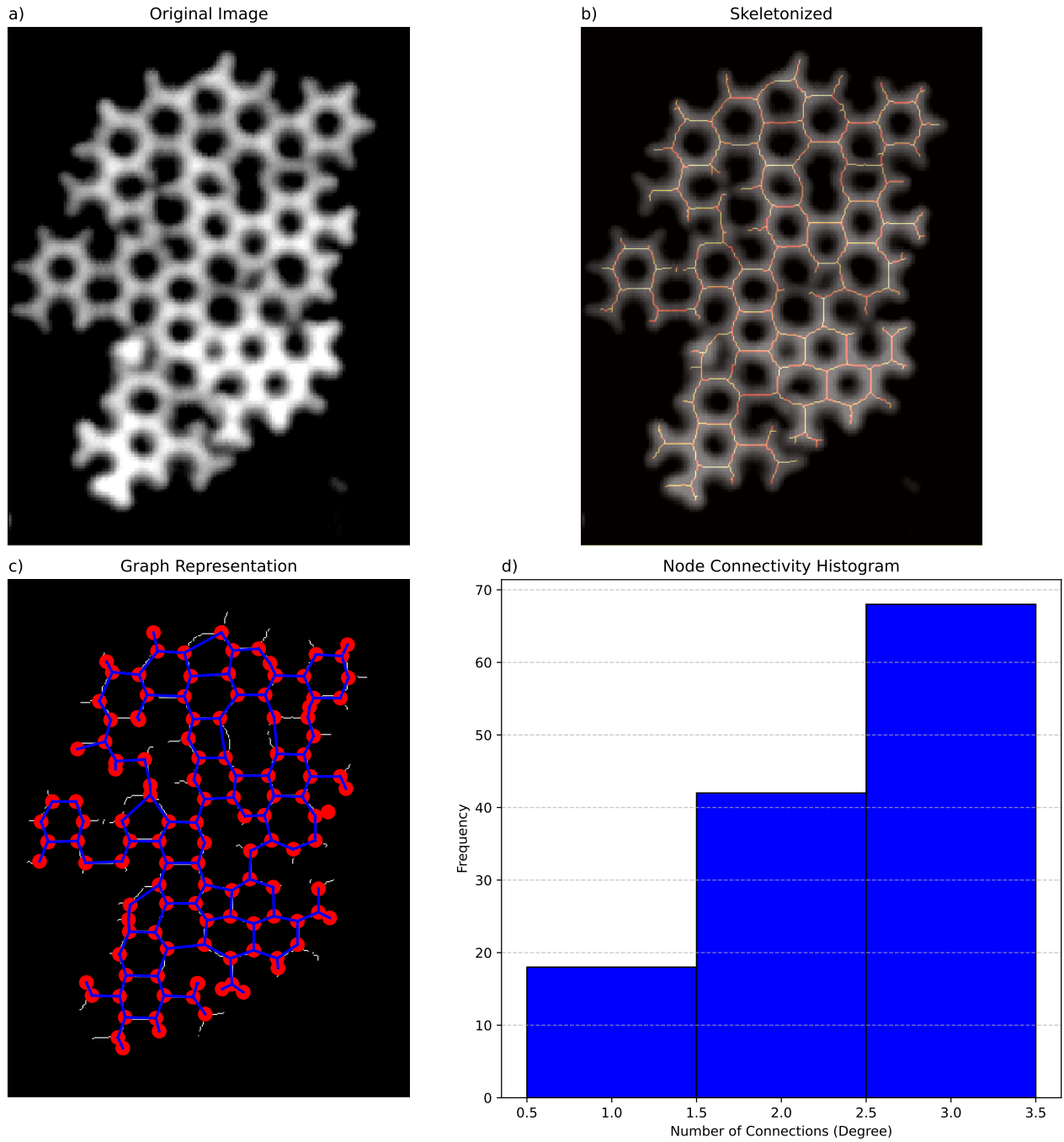


Figure 7: a) A single island identified from an STM image, corrected for background and gradient. b) Skeletonized island for further analysis of node connectivity. c) Skeletonized island with nodes and bridges identified. d) Final connectivity histogram for Wasserstein analysis applications.

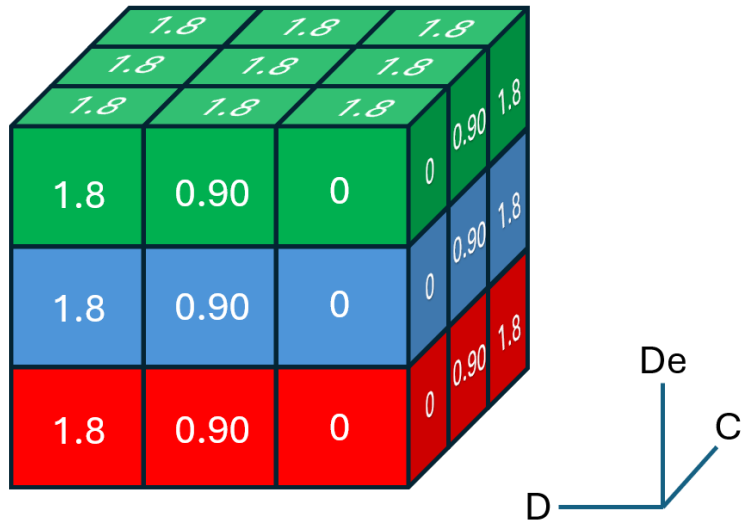


Figure 8: Parameter space for simulation energies (the actual simulations took 6 steps in each direction, shown in Fig. 9). The x , y , and z directions correspond to diffusion, coupling, and dehalogenation energies, respectively, in eV.

The simulations required to fill out the parameter space took six days to complete, running in parallel. This timeframe is expected for higher energies since the likelihood of performing an action on any given step becomes extremely small. This inhibits the diffusing monomer from coupling quickly, which is demonstrated in the exponentially increasing simulation times. Unfortunately, increasing the computational efficiency of the program will not necessarily result in reduced simulation times as the majority of the simulation time is tied to the action probabilities. In order to extend the maximum range of energies, much more computational time will be required or a fundamental change in the simulation procedure needs to be made. The simulated polymer islands were then compared to an experimental STM image in order to conduct Wasserstein analysis across the parameter space (Fig. 9). Based on the Wasserstein analysis for varying dehalogenation energies (rotation energy set constant at 0), there does not appear to be significant correlation between each of the dehalogenation steps, implying that changing the dehalogenation energy has important contributions to polymer formation.

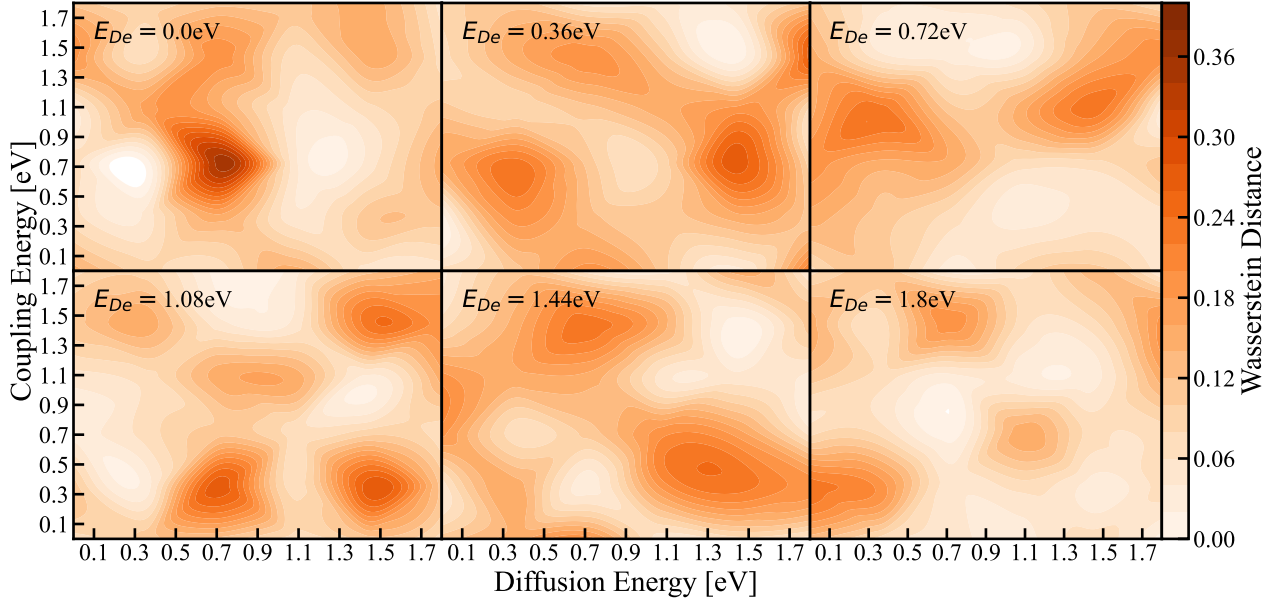


Figure 9: Wasserstein comparison of simulations at various energies with the polymer island shown in Figure 7. Rotation energy was taken as a constant throughout the simulation series because it does not seem to have a large impact on the Wasserstein minimum [7].

In this case, the absolute minima across all dehalogenation steps occurred at energies $E_D = 0.27\text{eV}$, $E_C = 0.69\text{eV}$, $E_{De} = 0.00\text{eV}$. While this energy generates the lowest Wasserstein distance after the contour interpolation, there are also several other combinations that produce very low distances: $E_D = 0.72\text{eV}$, $E_C = 0.72\text{eV}$, $E_{De} = 1.80\text{eV}$, $E_D = 0.71\text{eV}$, $E_C = 0.85\text{eV}$, $E_{De} = 1.80\text{eV}$, and $E_D = 0.85\text{eV}$, $E_C = 1.80\text{eV}$, $E_{De} = 1.08\text{eV}$. This range of possible combinations for energies that produce close matches to the experimental island implies that there is likely more parameters involved in the system that are not being taken into account. When applying the energies corresponding to the absolute minimum, the resulting simulated polymer has node connectivity almost exactly matching the experimental observations (Fig. 10). Importantly, the only parameter used to determine the likeness of a simulated polymer to the experimental image is the node connectivity, so the results of the simulation can be accepted if the histograms match. Unfortunately, as one can observe in the optimized simulated polymer, the “branchiness” is visually more extreme and extended than the experimental image,

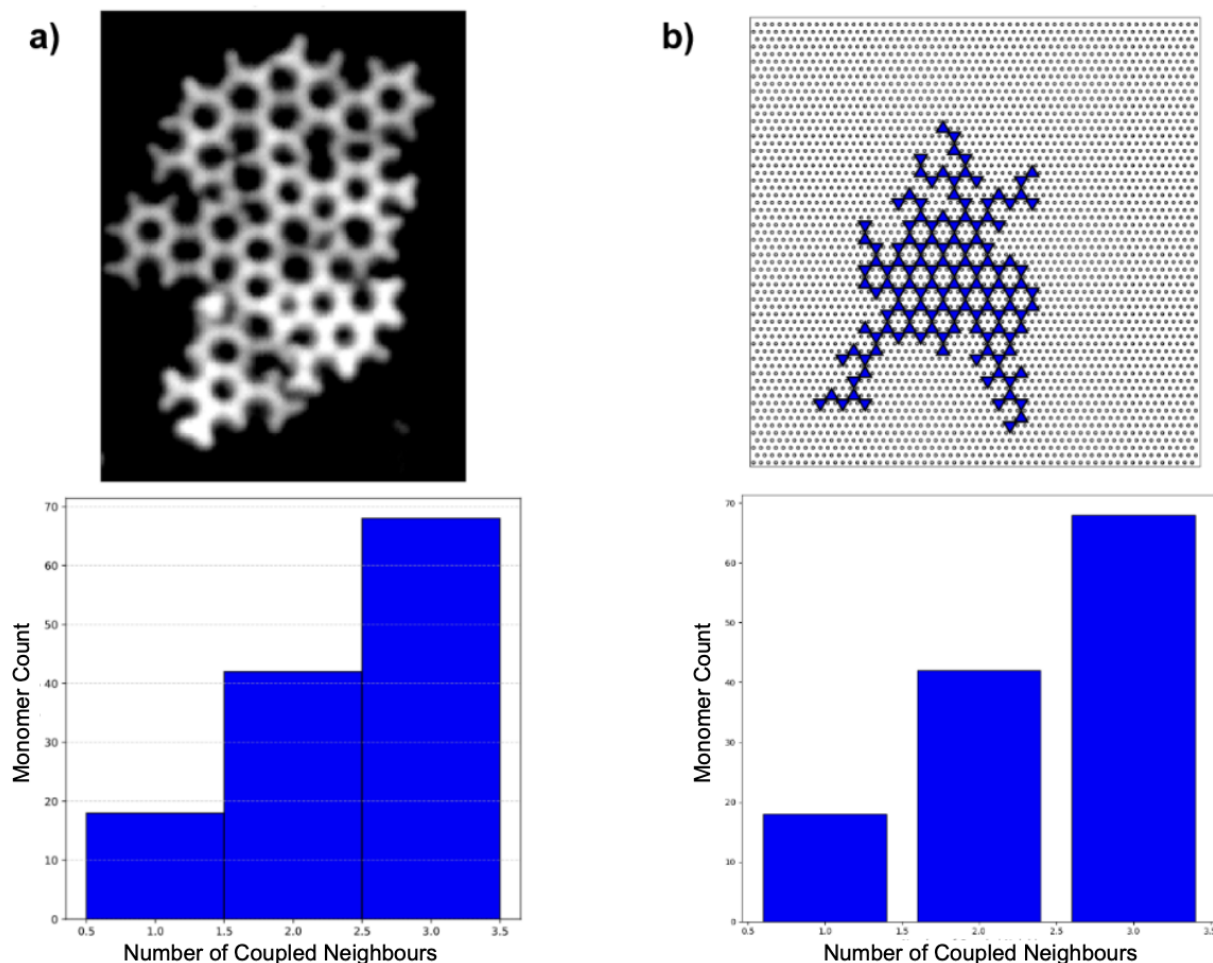


Figure 10: a) Experimentally observed polymer with analyzed connectivity histogram. b) Simulated polymer using the optimized energies from Wasserstein analysis along with the node connectivity.

implying a need for additional parameters to further constrain the structure of the polymers, potentially the radius of gyration, which is already a byproduct of the simulation analysis. As S. Briesenick [7] found, the diffusion and rotation energies do converge to an absolute minimum over several rotation energies. The energy range that are considered in our simulations are larger than those analyzed by Briesenick, so it is possible that additional minima are uncovered by expanding the range. More likely, however, is the difference between simulation techniques used. The Wasserstein distances shown in Figure 9 were found by applying the stochastic simulation process, while Briesenick used a kMC process. Implementation of dehalogenation into the kMC algorithm proved to cause significant problems with the rate

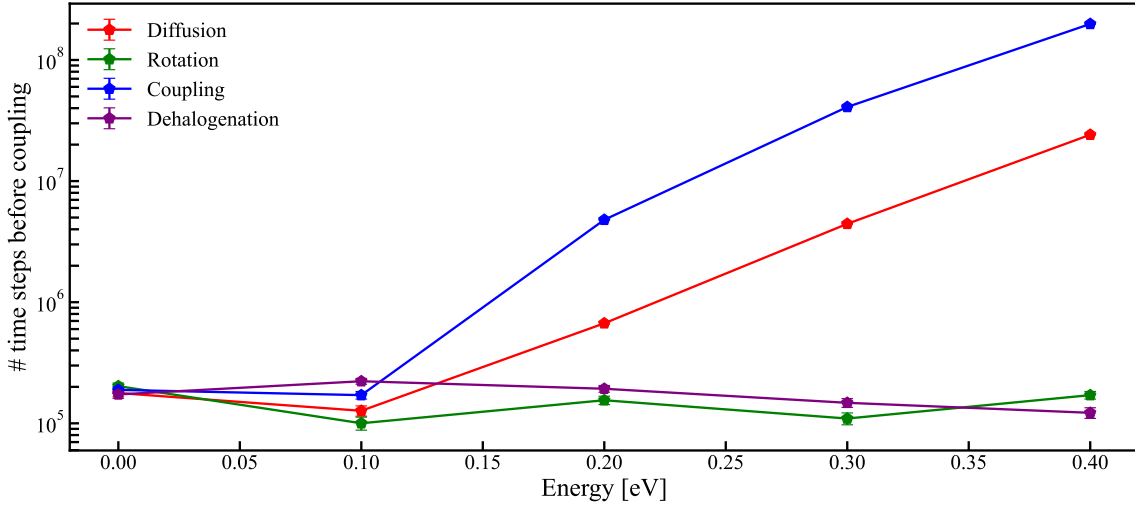


Figure 11: Number of time required for a single coupling event as a function of the energy passed to a specific parameter (diffusion, rotation, coupling, or dehalogenation) within the kinetic monte carlo simulation algorithm.

allocation of the program. Since the chosen event for a step is based on the relative probabilities for each action, adding a entirely new action reduces the acceptable range of energies due to the reduction in individual event probability simply from the existence of new options. When applying dehalogenation in the kMC algorithm, we observe extreme simulation times when the energy of any single parameter is further than 0.3eV from the others. The simulation times increased so much that no polymers were formed after three days of processing with the energies $E_D = 0.4\text{eV}$, $E_C = 0.7\text{eV}$, $E_R = 0.4\text{eV}$, and $E_{De} = 0.4\text{eV}$, a simulation that only required seconds using the stochastic algorithm. This problem was observed without the dehalogenation energies implemented in the MC program, but the time penalty for large disparities (larger than 0.3eV) in energy required at most one day for a single simulation. Instead of utilizing a similar energy range simulation menu to the stochastic algorithm, the relative increase in simulation time for varying different energy settings was observed (Fig. 11). One can observe an obvious exponential increase in the amount of added simulation time in when diffusion and coupling energies are increased, with relatively no change in the time per coupling event for increasing rotation and dehalogenation energies. The lack of increase in simulation time deriving from the rotation energy is reinforced by the Briesenick's findings [7], but the behavior from the increasing dehalogenation energy is contrary to the findings from

the stochastic simulations shown in Figure 9. This implies that there is a fundamental difference in the data output by the KMC and stochastic simulation algorithms, which will need to be examined in the future. The efficacy of the kMC simulation technique hinges on the speed of the simulated results, since it is unreasonable to wait for many days for a single Wasserstein distance corresponding to one set of energies. It is likely that some generalized calibration will need to be developed in order to enable the addition of more parameters to the simulation.

5 Conclusion and Future Outlook

Using the simulation workflow described in this project, we have demonstrated that analysis of experimental polymer islands is possible within relatively short time frames, using the stochastic simulation algorithm. The analysis may be expanded to include multiple island regions by incorporating wandering and step defects, which have also been implemented into the program. Despite these successes, the efficacy of kinetic Monte Carlo remains in question. While there has been quite a lot of success using kMC in reduced parameter spaces, attempting to conduct simulations over large energy ranges and with many possible actions appears to hinder the algorithm significantly. In the future, it will be necessary to resolve these issues in order to produce a robust kMC simulation algorithm.

6 Acknowledgments

A special thanks to our supervisor Peter Grutter, graduate student (now Ph.D student) Simon Briesenick, and postdoctoral fellow Wyatt Behn for their continuous and extensive support throughout this project.

References

- [1] V. B. Mbayachi, E. Ndayiragije, T. Sammani, S. Taj, E. R. Mbuta, and A. ullah khan, “Graphene synthesis, characterization and its applications: A

- review,” *Results in Chemistry*, vol. 3, p. 100163, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2211715621000680> 1
- [2] M. Lackinger, “Surface-assisted ullmann coupling,” *Chem. Commun.*, vol. 53, pp. 7872–7885, 2017. [Online]. Available: <http://dx.doi.org/10.1039/C7CC03402D> 2
- [3] J. Eichhorn, D. Nieckarz, O. Ochs, D. Samanta, M. Schmittl, P. J. Szabelski, and M. Lackinger, “On-surface ullmann coupling: The influence of kinetic reaction parameters on the morphology and quality of covalent networks,” *ACS Nano*, vol. 8, no. 8, pp. 7880–7889, 2014, pMID: 25036422. [Online]. Available: <https://doi.org/10.1021/nn501567p> 2
- [4] M. Bieri, M.-T. Nguyen, O. Gröning, J. Cai, M. Treier, K. Aït-Mansour, P. Ruffieux, C. A. Pignedoli, D. Passerone, M. Kastler, K. Müllen, and R. Fasel, “Two-dimensional polymer formation on surfaces: Insight into the roles of precursor mobility and reactivity,” *Journal of the American Chemical Society*, vol. 132, no. 46, pp. 16 669–16 676, 2010, pMID: 21043454. [Online]. Available: <https://doi.org/10.1021/ja107947z> 2
- [5] F. D. Pai Li, “Origin of herringbone reconstruction of au(111) surface at the atomic scale,” *Science Advances*, 2022. 7
- [6] S. Berg, D. Kutra, T. Kroeger, C. N. Straehle, B. X. Kausler, C. Haubold, M. Schiegg, J. Ales, T. Beier, M. Rudy, K. Eren, J. I. Cervantes, B. Xu, F. Beuttenmueller, A. Wolny, C. Zhang, U. Koethe, F. A. Hamprecht, and A. Kreshuk, “ilastik: interactive machine learning for (bio)image analysis,” *Nature Methods*, Sep. 2019. [Online]. Available: <https://doi.org/10.1038/s41592-019-0582-9> 7
- [7] S. Briesenick, “Kinetic and structural insights into the growth of two- dimensional conjugated polymers with scanning probe microscopy,” Master’s thesis, McGill University, 2024. 7, 9, 13, 16, 17, 18
- [8] R. Rangdee and P. Chatrathorn, “Effects of the ehrlich–schwoebel potential barrier on the wolf–villain model simulations for thin film growth,” *Surface*

- Science*, vol. 600, no. 4, pp. 914–920, 2006. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0039602805013701> 8
- [9] A. Ramdas, N. Garcia, and M. Cuturi, “On wasserstein two sample testing and related families of nonparametric tests,” 2015. [Online]. Available: <https://arxiv.org/abs/1509.02237> 13
- [10] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, Í. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python,” *Nature Methods*, vol. 17, pp. 261–272, 2020. 13