Summer Research Internship at EPFL

Chemiscope and Alchemical Kernels

Laboratory of Computational Science and Modelling
École Polytechnique Fédérale de Lausanne

Professor: Michele Ceriotti
Supervisor: Guillaume Fraux

Jakub Lála

Date of submission: 05 October, 2021
Word Count: 2292

1 Abstract

In the summer of 2020, I have spent 10 weeks working in Michele's Ceriotti COSMO lab at EPFL, Switzerland. The aim was to introduce myself to academic environment in-person and see whether I should be considering a future career at a university or a research facility. The placement was split into two main components - working on a visualisation tool called *Chemiscope* and coding a PyTorch model to express atomic representations in a computationally reduced form. After implementing several features and bugfixing the website version of Chemiscope, I have integrated it into Jupyter Notebooks as a widget to streamline its use. In the second part of the internship, I have successfully reproduced a feature optimisation technique previously suggested in the lab using the concept of alchemical kernels, where physical elements are represented as linear combinations of several pseudo-elements, reducing the amount of elemental information necessary for structural representation. Apart from that, I have improved my understanding of machine learning methods, web development, atomistic representations and advanced sampling methods in materials modelling.

2 Context

Laboratory of Computational Science and Modelling (COSMO) at EPFL is a research group led by Prof Michele Ceriotti. Their diverse research involves topics such as physics-inspired machine learning, nuclear quantum effects and path integrals, or dimensionality reduction and clustering. As a group heavily invested in computation, they put great effort into building various software tools for their own endeavours as well as other researchers [1]. For instance `librascal` [2] is a Python library to create atomic representations as the Smooth Atomic Overlap of Positions (SOAP) [3], whilst `Chemiscope` [4] is a website for data visualisation.

Having done a UROP with Dr Stefano Angiolleti-Uberti in the summer of 2020 on the topic of Monte Carlo DNA origami simulations, I knew I might want to pursue a career in computational modelling. Thankfully, Stefano recommended many professors, which then led me to landing a placement position in Michele's group, who is Stefano's former friend from university. Michele stood out primarily because of the cutting-edge research in applying machine learning in the world of computational materials science.

One of the main goals was to experience the academic research experience in-person, in order to evaluate whether I should later on pursue any further postgraduate studies. I also wanted to get my hands on machine learning to diversify my portfolio of transferable skills. Also, once I knew the content of the internship, I was very eager to get involved in the Chemiscope website, as it gave me a reasonably meaningful excuse to start learning web development. During the months prior to the internship, I had actually spent many hours going through a very detailed web development online course. Despite this web development part, I still wanted to explore the scientific aspect of the state-of-the-art research taking place at COSMO.

In terms of some general aims, I was looking forward to finally be working in a team on a meaningful and somewhat impactful project. Although research might not have such an explicit and clear impact, it is still very much more useful than sitting in lectures, studying for exams or collaborating on group coursework in the university bubble. As I am also generally interested in some sort of management since my involvement in the Year 3 Design Study as a Director, I have been intrigued by figuring out the mechanics behind in-person as well online teamwork collaboration.

Lastly, finding an internship position in Switzerland was also somewhat purposeful, as spending several months in a different country can lead to great cultural and personal insights. Therefore, I considered the placement to also be a learning experience in terms of living alone and abroad, after more than a year spent back home due to the COVID-19 pandemic.

3 The Placement

The placement was divided into two main sections and was done primarily under the supervision of one of the PostDocs, Guillaume Fraux. During the first five weeks, I was primarily focused on Chemiscope, whilst for the ending of the internship I directed my attention to creating an optimization PyTorch model using alchemical kernels.

3.1 Chemiscope

Chemiscope [5], shown in Figure 1 is a data visualisation app/website that is able to display both a map of properties and a structure of interest simultaneously. The user is thus able to easily browse through all

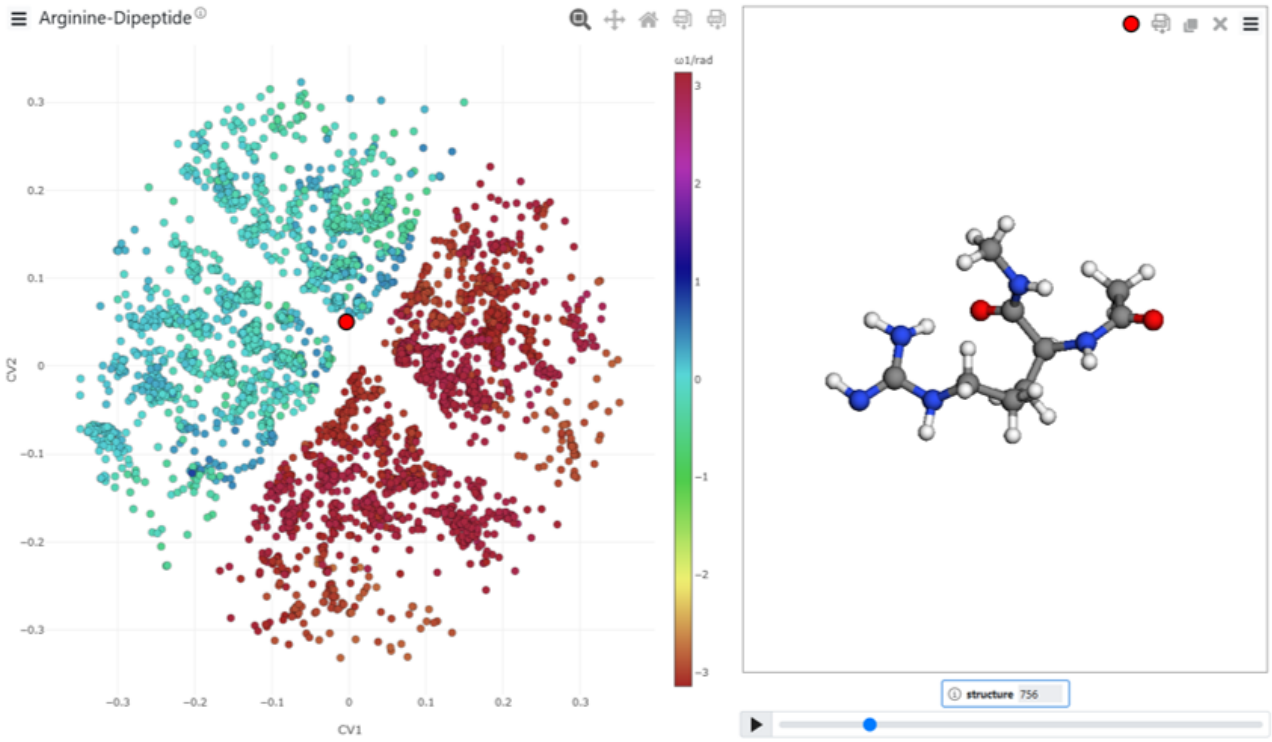


Figure 1: Chemiscope Website/App. On the left, the property map displays the properties of all the structures in the dataset, where the higlighted red dot then corresponds to the structure shwon on the right in the structure viewer. The dataset visualized is for Arginine-Dipeptide.

the structure candidates, looking at their relevant properties in the map, but also their structure. This can then also be extended to properties of individual atomic environments instead of the complete structure.

After the first days of extending my web development knowledge and getting used to working collaboratively on GitHub, I began contributing to the app by fixing the simpler issues posted on the repository. I was primarily coding in `TypeScript` (TS), which is built on top of `JavaScript` (JS). TS is a more readable, structured and less prone to error version of JS, as it allows for optional static typing that checks all the datatype manipulation in the IDE. That means that one can fix most of the type errors during coding, leading to a better performance during execution. Apart from that I was also coding in `HTML` and `CSS`, which define the architecture and the styling of the website respectively. The first finished contributions were either bugfixes or new features, such as adding download buttons, improving the functionality of the property map settings, or unifying and clarifying the user interface design. These contributions are listed in Table 1. In the third week, me and my supervisor organised a three-day hackathon to recruit more people from the lab to work on the project. During that time, I had implemented a completely new testing framework called Karma (Figure 2), as the previous framework did not allow us to properly utilise automatic testing for the app.

During the next couple of weeks, I have spent creating a Jupyter Notebook widget integration of the Chemiscope app. Before using Chemiscope on the website, one has to use the `chemiscope` Python library to convert his data into a specific JSON format. Having a Jupyter Notebook widget thus allows one to streamline that process and directly open up the visualisation app in the notebook without needing the website. Figure 3 shows the widget, whilst Figure 4 shows the idea behind the workflow shortcut. Apart from the install integration of the widget into the Python package, I was fully in charge of this project. I had to primarily ensure that the JavaScript front-end of the notebook, where the Chemiscope app instantiates itself, is able to communicate with the Python back-end of the `chemiscope` package. Afterwards, my work focused on bugfixing the layout and features that did not properly translate from the original Chemiscope app into the widget. Note that there was not enough time to make this extension work in Jupyter Lab.

3.2 Alchemical Kernels

For the second part of the internship, my job was to create a PyTorch model that would reproduce the work of one of Michele's previous papers about alchemical kernels [6]. The idea stems from optimizing the features that describe the structures (e.g. SOAP) to improve the learning rate of a machine learning

```
describe('MapOptions', () => {
    before(() => {
        // store karma's default HTML
        KARMA_INSERTED_HTML = document.body.innerHTML;
    });

    it('has a unique id in the page', () => {
        const root = document.createElement('div');

        const guid = 'this-is-my-id' as GUID;
        const options = new MapOptions(root, guid, DUMMY_PROPERTIES, DUMMY_CALLBACK);
        traverseDOM(document.body, (element) => {
            if (element.id) {
                assert(element.id.includes(guid));
            }
        });
        options.remove();
    });
});
```

Figure 2: Example of a unit test in the Karma testing framework

Table 1: My approved pull requests on Chemiscope's GitHub [4]

Pull Request ID	Date of Merge	Description
139	June 15	Added an option to download the property map as an SVG file
146	June 15	Added a warning when negative values are discarded once the map property is turned to logarithmic scale
153	June 28	Added a zoom reset feature on the structure viewer
154	July 5	Added an option to download the structure in the viewer as a PNG file
156	July 5	Improved the map settings (functionality and user interface design)
157	July 6	Added icons for the property table and the dataset title
161	July 7	First preparations for the Jupyter Notebook widget integration
173	July 21	Integrated of Karma testing framework
175	July 9	Added a small functioanlity feature in map settings
177	July 30	Added unit tests for map settings
178	July 13	Fixed a bug related to switching between structure and environment mode
179	July 13	Further preparations for the Jupyter Notebook widget integration
181	June 15	Added a configurable option for the maximum amount of viewers of the structures
183	September 3	Jupyter Notebook widget

model. This can be done in various fashions, but the approach we explored was the alchemical one, where the information about elements in a dataset is reduced based on a low amount of pseudo-elements. It is a sort of dimensionality reduction, where each element is then described as a linear combination of the pseudo-elements.

In the original paper, the authors had to use a complicated trick to define the loss function for the optimization algorithm to find the coupling parameters. I was able to code a PyTorch model, shown in Figure 5, that optimizes these directly without needing the trick. Firstly, I had to decide on the type of optimizer to use, which was fairly arbitrary as none of them clearly outperformed the other, as shown in Figure 6. After choosing to optimize with Stochastic Gradient Descent (SGD), the model's loss was not converging as expected (shown in Figure 7a). Only on the final day were we able to make it work (shown in Figure 7b) by coupling the learning of the weights and the coupling parameters instead of doing them separately, i.e. detaching the optimizing gradient backward propagation for the train path in Figure 5.

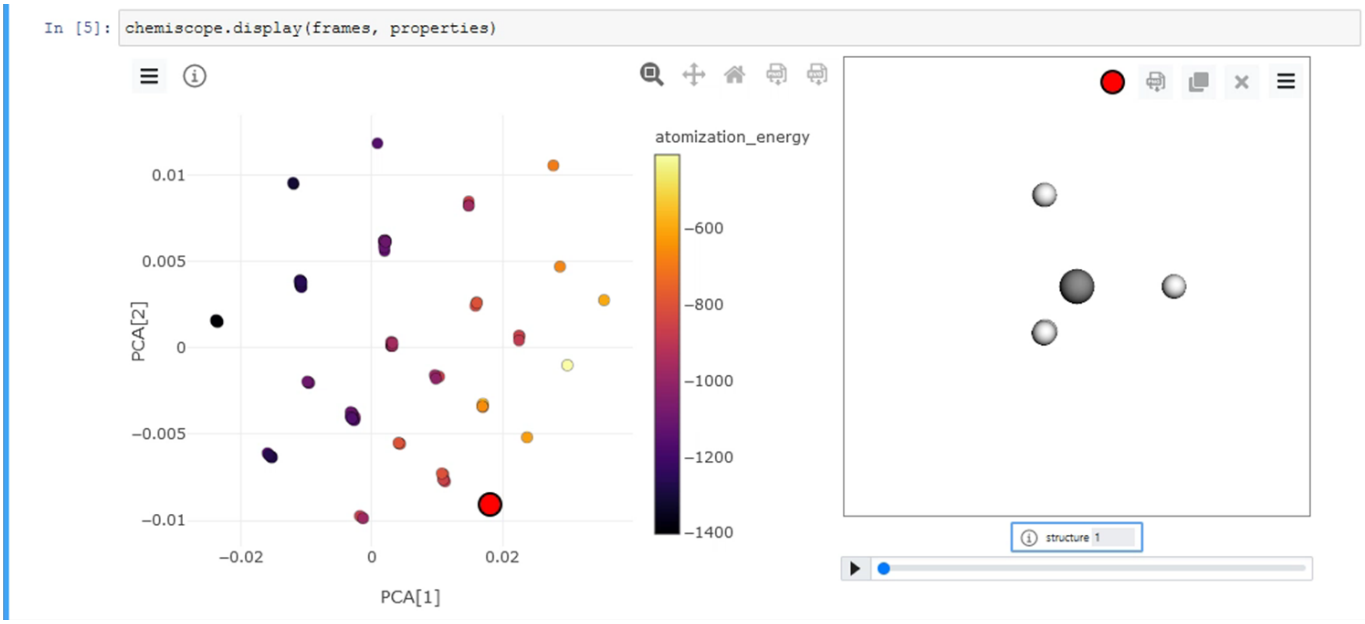


Figure 3: Chemiscope as a Jupyter Notebook Widget, where the `chemiscope.display()` functions instantiates Chemiscope directly in the output cell

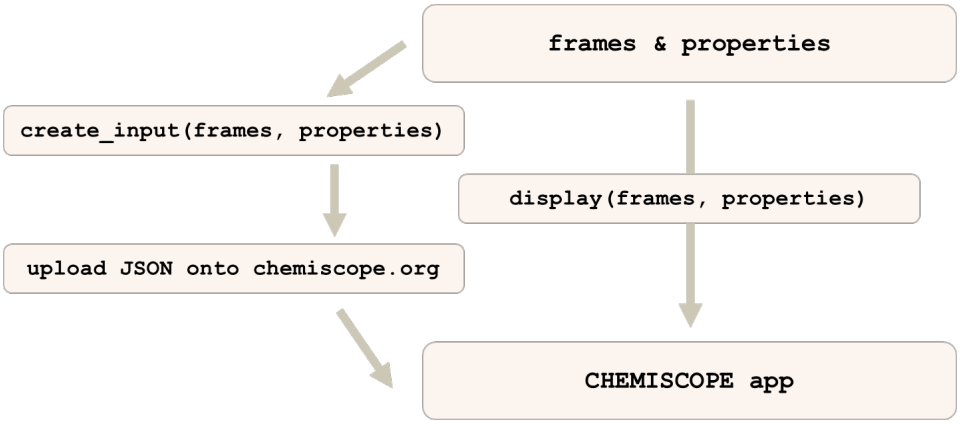


Figure 4: Streamlined workflow of using Chemiscope, where one does not need to upload the JSON input file onto the website, but instead directly instantiates Chemiscope as a widget in a Jupyter Notebook

All in all, Guillaume will now continue working on this model, implementing some more advanced feature optimization techniques to try and match the learning rate proposed in the original paper. Other people from the lab will also use the model in their research as they also deal with alchemical kernels.

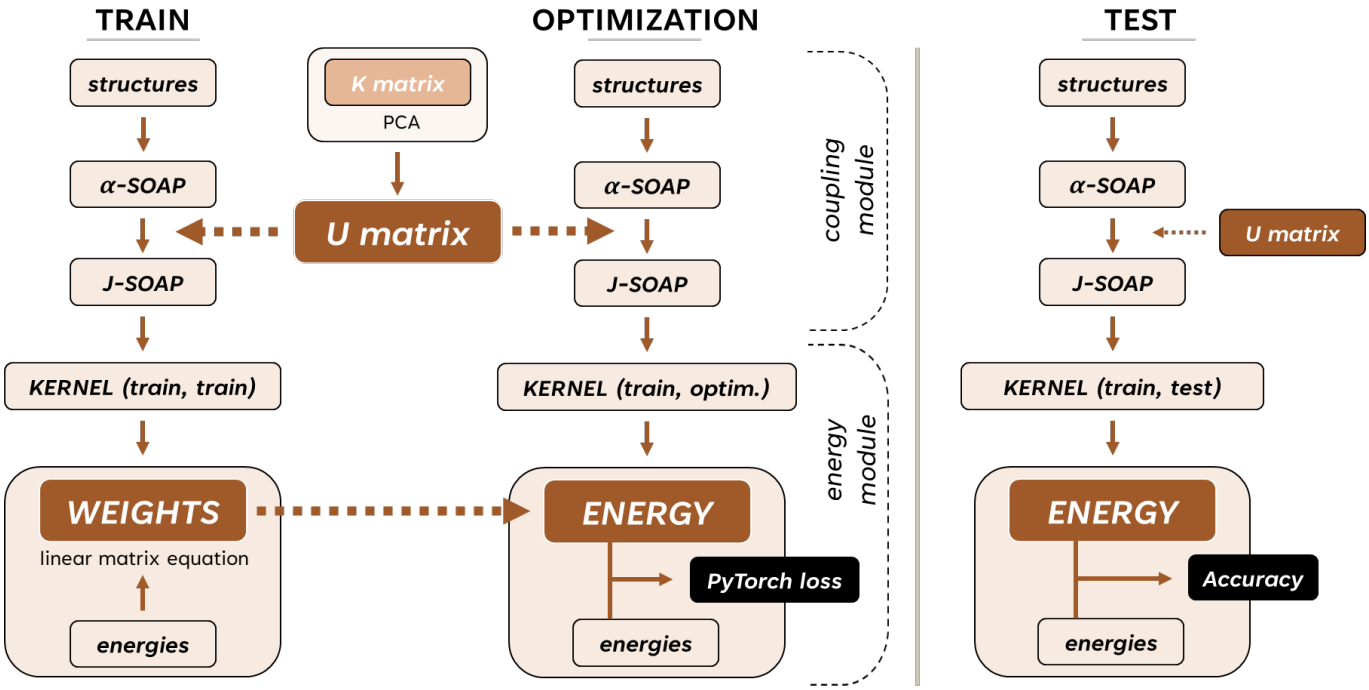


Figure 5: Schematic diagram of the PyTorch model created to optimize the coupling parameters (U matrix) between physical elements and pseudo-elements. These coupling parameters reduce the features of the structures to a smaller representation (α-SOAP to J-SOAP), which is then used to learn the kernel ridge regression weights to be then able to predict the structure energies. The elpasolite dataset used was separated into three sets - the train dataset is used to learn the weights to predict the energy, the optimization dataset is used to learn the coupling parameters and the test dataset validates the accuracy of the model

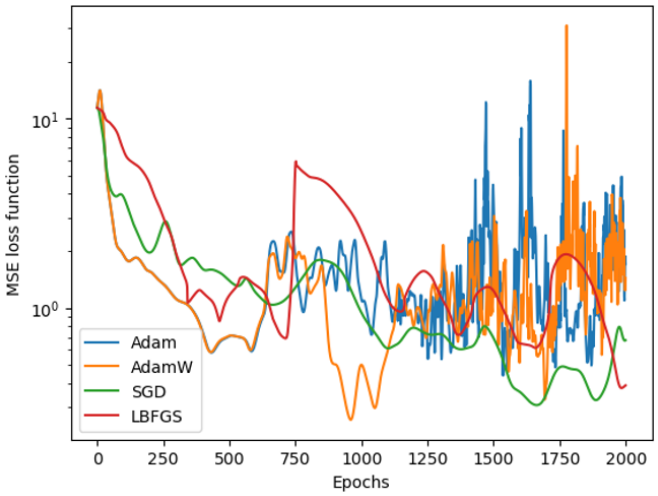
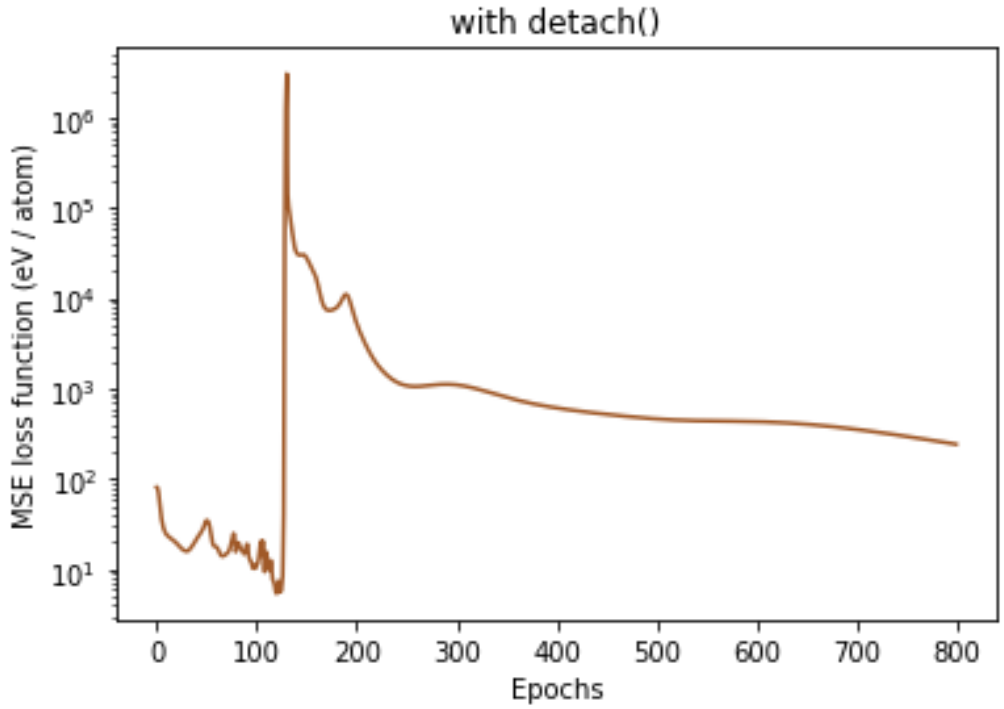
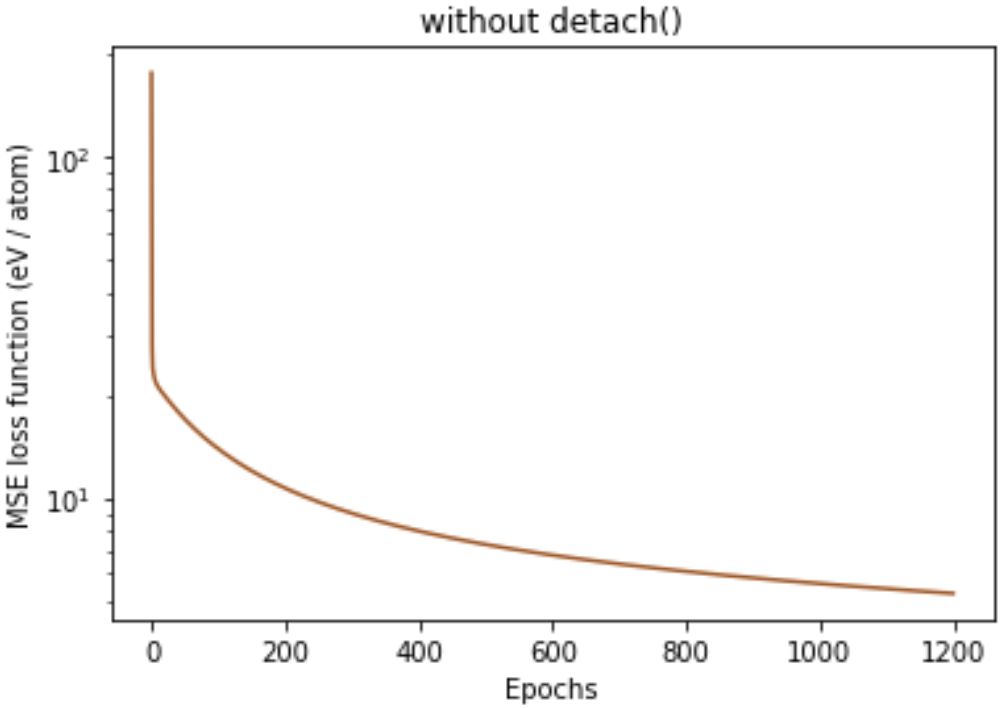


Figure 6: Evolution of the PyTorch model loss for different optimizers



(a) gradients detached from training back-propagation



(b) gradients back-propagate through the training path

Figure 7: Comparison of the PyTorch models, where **a)** does not back-propagate the gradients through the training path, whilst **b)** does back-propagate the gradients through both the optimization and the training paths in Figure 5. The plots show the evolution of the mean squared error loss of the predicted energy of a structure with respect to the actual energy.

4 Lessons Learnt

In terms of practical skills, I have become more accustomed to using GitHub for coding collaboration and Slack for team communication. I also improved my Python coding experience and developed a basic understanding of the PyTorch framework, which is gaining traction within the machine learning community. My web development skills (HTML, CSS, TypeScript) have also been consolidated and I am looking forward to applying these in future personal projects, or free-lancing jobs.

In terms of scientific knowledge, I have gained much more insight into the workings of atomistic representations and how clever maths used in SOAP allows us to describe structures in complete, unique and effective ways. I have also learned the basic idea of kernel ridge regression and how it can be applied to machine learning tasks. I had also the pleasure to visit some public PhD thesis defenses, where I further expanded my knowledge of deep learning methods, e.g. on how to apply both supervised and un-supervised learning methods in conjunction to produce useful conclusions [7]. Moreover, I have audited one of Michele's Master's summer courses, where I polished my understanding of advanced Statistical Mechanics and sampling methods in Monte Carlo simulations.

Apart from that, I have learned that having a supervisor in the same office is extremely beneficial to get the ball rolling. In the first weeks, Guillaume was able to help me with the most basic questions about web development, which accelerated my contribution progress. Furthermore, I have understood that having a robust network of colleagues and people in the field is extremely valuable. Not only was this internship recommended by my former supervisor, but Michele suggested my current Master's thesis supervisor at MIT.

Lastly, I am probably not going to stay in academia for the rest of my life, but I might be considering some attractive PhD positions in the world - either in the US, or in Switzerland. I have made this decision primarily from my long discussions with PhDs and PostDocs at EPFL. Nevertheless, the industry, the entrepreneurship, or the start-up worlds are still alluring me, as I do not want to spend long hours sitting behind a computer coding material modelling software.

Acknowledgement

I would like to express my special thanks of gratitude to my hosting professor Michele Ceriotti, as he was the one that chose my name from the immense pile of intern applicants, and gave me the opportunity to spend a beautiful summer at EPFL. Secondly, I need to thank my direct supervisor Guillaume Fraux, who was extremely kind and helpful, and although he went on several vacations during my internship, he was always quick to respond via Slack. He also proved to be a strong emotional support during my minor mental breakdown on the second day of the job, where my mind got consumed by the imposter syndrome. Thirdly, I need to thank my former supervisor Stefano Angioletti-Uberti, who recommended Michele as a rising star in machine learning. Lastly, I want to thank all the people at EPFL that organised this research internship scheme, and all the people I met during my time in Switzerland as you were truly making my summer experience all the better.

5 References

[1] *Laboratory of computational science and modeling*. URL: https://github.com/cosmo-epfl.

[2] Cosmo-Epfl. *Cosmo-EPFL/librascal: A scalable and versatile library to generate representations for atomic-scale learning*. URL: https://github.com/cosmo-epfl/librascal.

[3] Albert P. Bartók, Risi Kondor, and Gábor Csányi. "On representing chemical environments". In: *Physical Review B* 87.18 (May 2013). ISSN: 1550-235X. DOI: 10.1103/physrevb.87.184115. URL: http://dx.doi.org/10.1103/PhysRevB.87.184115.

[4] Cosmo-Epfl. *Cosmo-Epfl/Chemiscope: An interactive structure/property explorer for materials and molecules*. URL: https://github.com/cosmo-epfl/chemiscope.

[5] URL: https://chemiscope.org/.

[6] Michael J. Willatt, Félix Musil, and Michele Ceriotti. "Feature optimization for atomistic machine learning yields a data-driven construction of the periodic table of the elements". In: *Phys. Chem. Chem. Phys.* 20 (47 2018), pp. 29661–29668. DOI: 10.1039/C8CP05921G. URL: http://dx.doi.org/10.1039/C8CP05921G.

[7] Benjamin Aaron Helfrecht. "Structure-Property Relationships in Complex Materials by Combining Supervised and Unsupervised Machine Learning". In: (2021), p. 154. DOI: 10.5075/epfl-thesis-9032. URL: http://infoscience.epfl.ch/record/287075.