Alma Mater Studiorum - Università di Bologna

DOTTORATO DI RICERCA IN
COMPUTER SCIENCE AND ENGINEERING

CICLO XXXV

Settore Concorsuale: 01/B1 - INFORMATICA
Settore Scientifico Disciplinare: INF/01 - INFORMATICA

Data-driven and data-oriented methods for materials science and technologies

Presentata da: Fabio Le Piane

Coordinatore Dottorato:
Ilaria Bartolini

Supervisore:
Mauro Gaspari

Co-Supervisore:
Mario Bravetti

Co-Supervisore:
Francesco Mercuri

Esame finale anno 2023

Abstract

The discovery of new materials and their functions have always been a fundamental component of technological progress. Nowadays, the quest for new materials is stronger than ever: sustainability, medicine, robotics and electronics are all key assets which depend on the ability to create specifically tailored materials.
However, designing materials with desired properties is a difficult task, and the complexity of the discipline makes it difficult to identify general criteria. While scientists developed a set of best practices (often based on experience and expertise), this is still a trial-and-error process.
This becomes even more complex when dealing with advanced functional materials. Their properties depend on structural and morphological features, which in turn depend on fabrication procedures and environment, and subtle alterations leads to dramatically different results.
Because of this, materials modeling and design is one of the most prolific research fields. Many techniques and instruments are continuously developed to enable new possibilities, both in the experimental and computational realms. Scientists strive to enforce cutting-edge technologies in order to make progress. However, the field is strongly affected by unorganized file management, proliferation of custom data formats and storage procedures, both in experimental and computational research. Results are difficult to find, interpret and re-use, and a huge amount of time is spent interpreting and re-organizing data. This also strongly limit the application of data-driven and machine learning techniques.

This work introduces possible solutions to the problems described above. Specifically, it talks about developing features for specific classes of advanced materials and use them to train machine learning models and accelerate computational predictions for molecular compounds; developing method for organizing non homogeneous materials data; automate the process of using devices simulations to train machine learning models; dealing with scattered experimental data and use them to discover new patterns.

Contents

List of Figures

List of Tables

Introduction

Materials have always been the main engine of innovation: from tackling completely new problems to optimizing existing devices, discovering new materials is the main way to boost technological progress[29]. Our time, marked by the quest for green energy, is probably the pinnacle of this historical need for advanced materials; as a result, we now need to go beyond materials with just specifically tailored properties but we also need to engineer the life cycle of the same materials including the fabrication procedure.

Materials science, historically, has been one of the fields that has seen the strongest utilization of computational techniques and the best integration with experimental lines, taking advantage of the respective strengths of these two paradigms to propel discovery. However, this also led to more complexity, and in particular in the difficulty of merging information and data coming from the two realms to one another. Moreover, due to a lack of a motivation to unify and standardize data formats and processing procedures, this problem is further aggravated by the proliferation of different file formats, which are often specifically thought for a precise software and used for specific computational workflows or for collecting data coming from an experiment performed with a specific instrumentation.

For these reasons, we need to provide researchers with powerful tools for developing and pursuing their scientific questions without worrying about the way data is stored or imposing a way to format data that is dependent on the choice of a specific software or instrument stack. These tools are needed both for experimental and computational research, and the most crucial common

trait is the need for suitable data management and storage platforms that also empower the re-usability of the data produced in previous research as the starting ground for new activities.
This thesis targets applications based on specific classes of advanced materials, where structural features, morphology and functional properties are spread across a very broad range of dimensional scales. Despite the resulting multi-scale complexity, this variability is at the basis of several technological applications of advanced functional materials. This is for example the case of nanotechnology, where the peculiar properties of materials at the nanoscale are exploited in order to engineer advanced functionalities. In this work, we will address different research topics related to the field, from the development of innovative materials to their applications in advanced devices. The complexity of the relationship between morphology and composition of basic structural units, materials processing and fabrication conditions and environments, and resulting functional properties across broad dimensional ranges, however, introduces additional difficulties. As a result, the design and engineering of new materials targeted to specific functionalities and applications still constitutes a very challenging task. As stated previously, data-driven approaches can potentially boost research in the field, providing the scientific and technological basis for the development of predictive platforms and for the automation of complex processes. While this situation has been partially mitigated in some cases through the development of curated data platforms[65], the development of structured, data-driven approaches for specific classes of functional, multi-scale materials is still at the early stage. In the work discussed in this thesis, we considered both computational and experimental approaches to materials development as enablers of workflows for generating information, knowledge and data on materials. The integration between simulations and experiments is indeed a key element to advance the research on materials for applications in technology. Accordingly, we focused on three main aspects related to advanced and functional materials: materials multiscale structure and properties, materials fabrication and processing,

and materials applications. While these aspects involve different concepts and procedures depending on the specific investigation and/or development tools used (for example, materials modelling or materials characterization tools) many of the underline ideas, basic conceptual frameworks and issues are similar if not identical. We will elaborate more on their similarities and differences in the next chapters; in the rest of this introduction, we introduce the main ideas and methodologies that we faced during this work both for the computational and experimental realms.

In the following, we discuss a broad range of applications and demonstrators developed during the PhD research work, introducing potential solutions for addressing the issues mentioned above. We will also show examples of how putting the individual pieces together enables the automation of entire complex procedures, thus improving the overall throughput of research efforts. Along the thesis, we are going to tackle different specific problems:

Multi-scale materials features We show how to develop more general features for specific advanced functional materials, and in particular for molecular and nanoscale materials. In particular, we show how to deal with complex materials morphologies, developing features that are deeply linked to the knowledge about a scientific question. These features should be easily understandable for researchers, which could then enforce their existing knowledge of the physical problem in order to build the best solution possible; at the same time, those features must be easily processable by machines, leading to efficient computations, easier training of the actual machine learning models and more efficient predictions. This approach can for example lead to the development of lightweight software that is able to predict materials properties with accuracy comparable to that of computational simulations and at a fraction of the computational cost.

Data management We discuss about how to deal with data related to specific advanced materials classes in order to gather information about com-

plex materials applications (e.g., devices) and organizing data in the field, giving the fundamental pillars for the creation of a fully featured database for the field. This last endeavor has been pursued enforcing the power of semantic technologies, and in particular ontologies. To this end, we developed a domain ontology targeting molecular materials, a specific class of advanced materials. We introduced possible procedures to convert the existing plethora of data formats to a single, standardized one. The expressiveness of the resulting format is grounded on the knowledge representation resulting from ontology development. We also introduced a possible framework for automating different procedures related to materials R&D activities. This is based on the application of Problem Solving Methods, and the interplay between those methods and ontologies is a key asset in the creation of a common platform for the storage of results and the workflows that led to those results.

Simulation of devices based on advanced materials and automation We discuss about how to use the results of simulations of devices based on advanced materials, to train machine learning models. This leads again to huge performance improvements. However, dealing with phenomena and entities that arise at devices scales leads to a manifold of issues, which we tackled using solutions that are similar to those enforced in the study of materials properties at lower dimensional scales. Moreover, we used this work as a test-bed for developing automation tools for computational workflows, and we introduce tools used to automate the whole pipeline that goes from the definition of the scientific question to the actual trained model and the predicted properties.

Experimental materials data and pattern recognition We show to deal with scattered experimental data on advanced materials applications, how to process them and how to use them to train learning algorithms capable of qualitatively predict significant performance indicators.

The aforementioned activities shows prototypes of possible solutions for the limitations introduced above, and are meant to be the basics of future development aimed at integrating all those techniques into a unified set of tools. Other than standardization and re-usability, we hope that the integration of these different steps can help to overcome the difficulty of developing multiscale computational and experimental workflows. Moreover, we strove to develop software that are easy to use and to integrate into pre-existing solutions, in order to make data-driven technologies more attractive to researchers. We will show examples of that ease-of-use approach along the thesis.

The thesis is structured as follows: in chapter 1 we are going to introduce the whole scientific and technological context needed for understanding our work and contribution together with the state of the art regarding the application of ML and data science in general to the field of advanced materials development. In chapter 2, we introduce the work done on using machine learning techniques to automatically predict materials properties from computational data. In chapter 3, we go through the work done to create a domain ontology specifically tailored for molecular materials, also putting the emphasis on general data quality and, in particular, on the integration between data stemming from computational activities and those coming from real-world experiments. In chapter 4 we are going to illustrate the work done on examples of applications of advanced materials in electronic devices. In particular, we considered transistors based on organic electronic semiconductor materials, using computational data to train machine learning models that are able to extract hidden properties of devices. Future developments of this work include the realization of models that are able to predict properties that are difficult to access experimentally from devices. In chapter 5 we show an example of how we can use data on technological applications of advanced materials originating from historical and publicly available knowledge to develop a model for predicting and optimizing application performances.

Finally, chapter 6 will be devoted to the conclusions and possible future works.

Chapter 1

Scientific and technological context

Before going deeper into the analysis of the experimental and computational methods with whom we interacted, we have to introduce the general context and notions needed to describe the physics and chemistry that regulates the entities and phenomena at hand.

Historically, materials have always been a key enabler of new and revolutionary technologies, and now more than ever the ability to tailor materials for specific functions is the main driver for innovation and improvement[26]. Advanced materials are crucial for many of the most impacting social sectors like health, energy, mobility and housing. Moreover, they also come in contact with the general public through consumer goods like construction materials, cleaning and hygiene-related products, cosmetics and many more. In figure1.1 we can see an infographic showing in more detail the sectors that strongly rely on the development of advanced materials.

To cite the Materials 2030 Manifesto[29]:

> Materials, especially advanced materials, are the backbone and source of prosperity of an industrial society. In the context of the

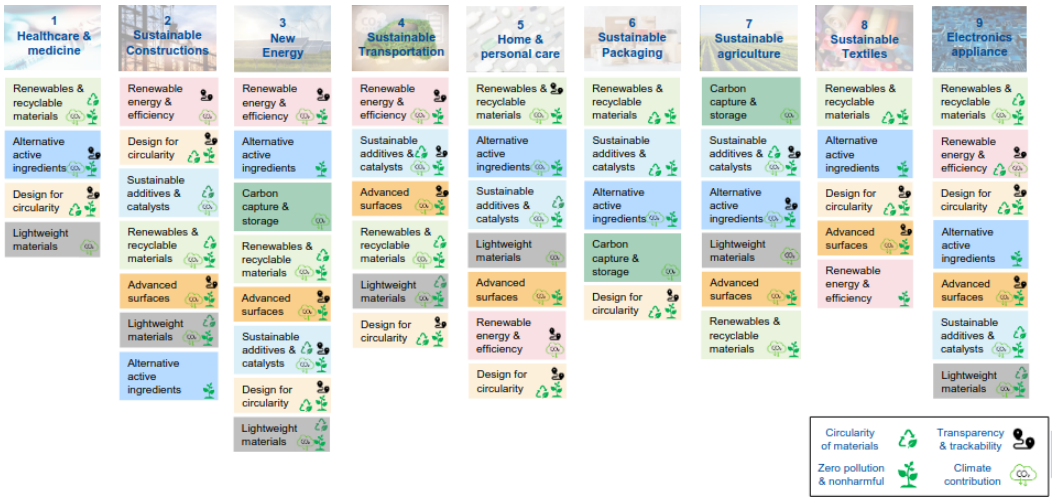


Figure 1.1: An infographic, showing nine crucial sectors for which advanced materials are a key enabler. Adapted from [29]

> radical transformational changes of the 21st century, it is precisely these advanced materials that will play a decisive role.

One of the key challenges for the future is the digitalisation of materials science research, and in particular the creation of a Common Digital Ecosystem[1]. Specific efforts are going to be made in stepping up the level of the data management technologies employed in the field in order to ensure availability, transparency and horizontal access to data, and this will lead to easier access to data-driven technologies[1] applied to the discovery and design of new materials and their applications.
The work done during these three years fit in this bigger scheme, as we strove to give researchers new tools to accelerate their work, to improve their ability to better understand the properties of the materials at hand and, ultimately, leading to the design of novel materials with improved functionality and properties.

One of the most interesting classes of materials for advanced applications in technologies is that of molecular and organic materials. This class of ma-

terials exhibit a broad range of structural and physico-chemical properties across a wide range of dimensional scales, from the molecular level to the bulk. As a result, molecular materials have found applications in several fields of technology and nanotechnology, where the peculiar behavior of materials at the nanoscale enables specific functionalities. In this chapter, we introduce the generalities of organic and molecular materials (and in particular their properties as semiconductors) and of the applications based on them, for example in the realization of advanced devices. Particular care will be given to advanced functional devices, such as transistors and solar cells, two macro-classes of devices that are the main focus of two of the following chapters, namely chapter 4 and chapter 5 respectively. We introduce fabrication techniques and experimental methods in general and computational methods. Then, we also discuss how the data are currently produces and stored in the materials science domain.

1.1 Molecular materials and organic semiconductors

Molecular materials constitute one of the most interesting classes of materials for the development of applications in technology. As such, molecular materials enable innovation in a very broad range of fields, including nanoelectronics, photoelectronics and photonics, quantum computing, energy and information storage, and several others. The basic building blocks of molecular materials are typically constituted by molecular units or sub-units. Consequently, one of the most typical features of molecular materials is the complex relationship between structure and morphology across a broad range of scales, from molecular structure to the nanoscale, and the resulting materials properties. Among molecular materials, organic semiconductors (OSs) are carbon-based materials usually based on molecular units or sub-units (small molecules or polymers), which exhibit properties suitable for being used in electronics as semiconductors. To fully understand the physical rea-

son for these properties it is important to analyse the electronic structure of the carbon atom and, in particular, the kind and number of bonds that it can form. Carbon has four electrons in the outer energy level (i.e., it can form four bonds with other atoms) and, more importantly, the carbon atom can hybridize in several forms. The concept of hybridization, introduced in 1931 by Linus Pauling, describes the linear combination between different atomic orbitals. In particular, carbon can form three different kinds of hybrid orbitals named sp, sp^2 and sp^3. These represent the combination of s (that are the ones that have a spherical symmetry) and p (that are the ones with symmetry shaped like two distinct spheres) orbitals. Let us consider as an example the sp^2 hybridization shown in Figure 1.2.

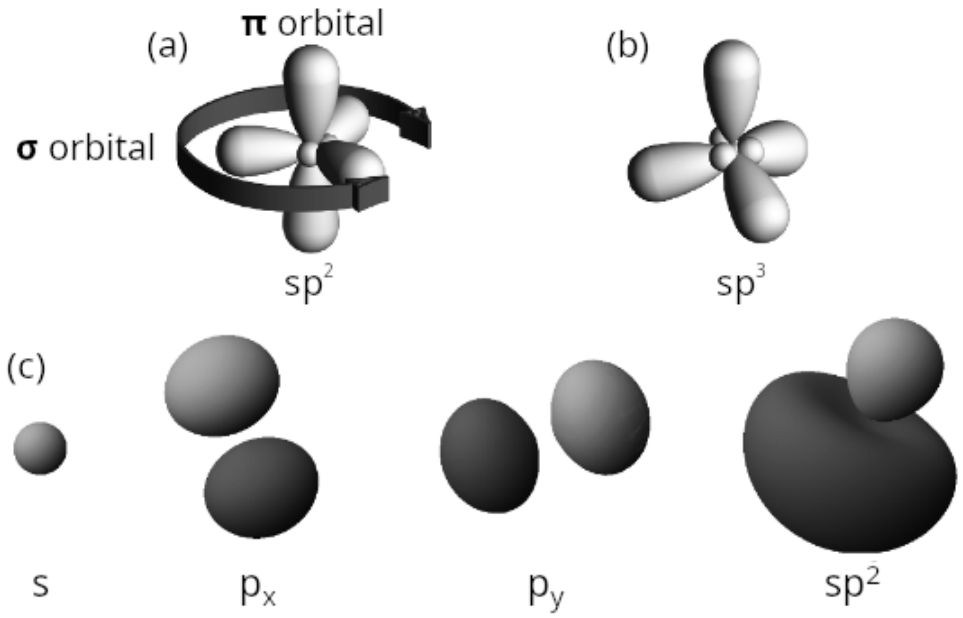


Figure 1.2: A figure depicting the geometry of the sp^2, and sp^3 orbital structures in (a) and (b). (c) shows the electron density probability surface calculated for the components of the sp^2 orbital together with one lobe of the hybrid orbital itself. This figure is adapted from [84]

This scenario gives birth two main type of (covalent) bonds:

- The two p orbitals (one on the x axis, one on the y axis), overlapping, form a bond called σ-bond
- The partial overlap of the two p orbitals on the axis z form the so called π-bond

Energetically speaking, the much larger overlap between the two sp^2 orbitals if compared with the two unhybridized $2p_z$ orbitals leads to a difference in strength between these two: the σ-bond is a stronger bond than the π-bond[84].
These two bonds give very different electronic properties to the material based on carbon atoms involving these kinds of bonding patterns: in the σ-bond, the involved electrons (called σ-electrons) are more localized, and because of this they do not have much freedom of movement; on the other hand, the π-bond leaves more freedom to its electrons (called π-electrons). For these reasons, π-bonds usually lend to better electric and electronic properties, while σ-bonds have stronger structural properties.
Usually, organic materials with semiconductor properties are formed by distinguishable units, linked one to another by π-bonds. Depending on the length of the chain formed by these bonds, we can classify materials into two groups, namely those made of small molecules (Fig 1.3) and polymers (Figure 1.4). The first have a very well-defined molecular characteristics (like molecular weight), while the latter are made of long-chain molecules, built with an indeterminate number of repeating building blocks[1]. Despite this intrinsic difference, these classes of compounds share many traits and similarities, in particular regarding their optical and electrical/electronic properties[38, 18].

Besides this qualitative description, we can also have a quantitative knowledge of these materials and phenomena through the theory of Molecular Orbitals (MOs)[37]. It states that we can describe the orbitals of complex

[1]Which are smaller molecular units themself.

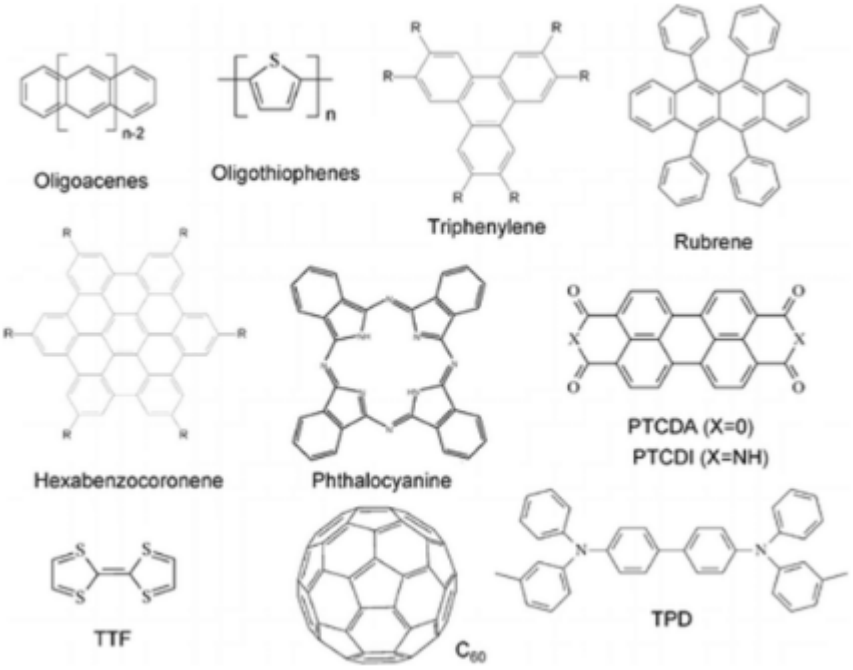


Figure 1.3: Chemical structure of some of the most studied organic small molecules. Adapted from [28]

molecules using linear combinations of the atomic orbitals of the single elements of the molecule itself. This method provides specific mathematical equations for describing the energetic structures of a molecular system, identifying two categories of energy levels. These are called bonding (π) and anti-bonding (π^*) orbitals.

Another important pair of concepts is that of HOMO (Highest Occupied Molecular Orbital) and LUMO (Lowest Unoccupied Molecular Orbital). These are, respectively, the outer occupied orbital and the lowest unoccupied one, which we can see as analogous to the valence band and the conduction band of a traditional silicon-based semiconductor. Moreover, the energy difference between HOMO and LUMO represents the energy gap of the semiconductor. It must be noted that, within the description provided by the aforementioned MOs theory, in most OSs the HOMO corresponds to the occupied π levels,

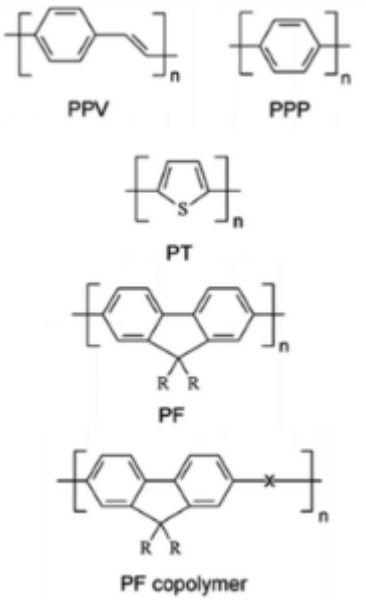


Figure 1.4: Chemical structure of some of the most studied polymers semiconductors. Adapted from [28]

while the LUMO corresponds to the unoccupied π^* levels.
To accurately describe the charge transport mechanism in organic semiconductors, we have to acknowledge the fact that, while the atoms of traditional inorganic semiconductors are linked by covalent bonds, organic molecules aggregates are kept together by the weaker Van der Waals force. For these reasons, this class of materials has lower conductivity, and we need a new model to describe their completely different charge transport mechanism.

1.1.1 Charge transport in Organic Semiconductors

In the previous section, we discussed that, while inorganic semiconductors usually exhibit a band transport mechanism (due to the delocalized states of the electrons), organic materials rely instead on Van der Waals interactions, with charge transport occurring between localized states. In particular, or-

ganic materials transport mechanism relies on the overlap between π orbitals; this implies that the electronic transport performance is strongly dependent on the structural characteristic of the organic material, and in particular on how molecular units are arranged in space one with regard to another. It follows that the degree of order at the molecular level in OS materials plays a fundamental role in determining the charge transport properties. Accordingly, charge transport in OSs can be described using the band formalism for very ordered materials morphologies (like organic single crystals, that tend to be very ordered systems with long-range chains) or using the hopping model otherwise. Amorphous OSs fall into this second, less-ordered class and will be discussed in the next chapters.
Even if the full details of the hopping mechanism in amorphous materials are still partially unknown, several formalisms are already presented in various pieces of literature, as those illustrated in [28, 39, 165, 59, 156]. The following section will introduce the most used ones.

Charge Hopping

Firstly introduced in [119] and [27], and then revisited in [116], this model takes in consideration the difference between an ordered system (where the electrons can move freely between delocalized states) and disordered or amorphous solids, where charge transport occurs through hopping between localized states. To find the best suited model, we need to observe the relation between the mobility and the temperature, looking at their mutual variations. It is known that, when using the band-like model, increasing temperature leads to a drop in mobility. This relation is visible in the plot in figure 1.5.

On the other hand, materials relying on hopping for charge transport experience an increase in mobility when temperature rises. More specifically, a model based on hopping sees mobility as proportional to the transition rate between different states.

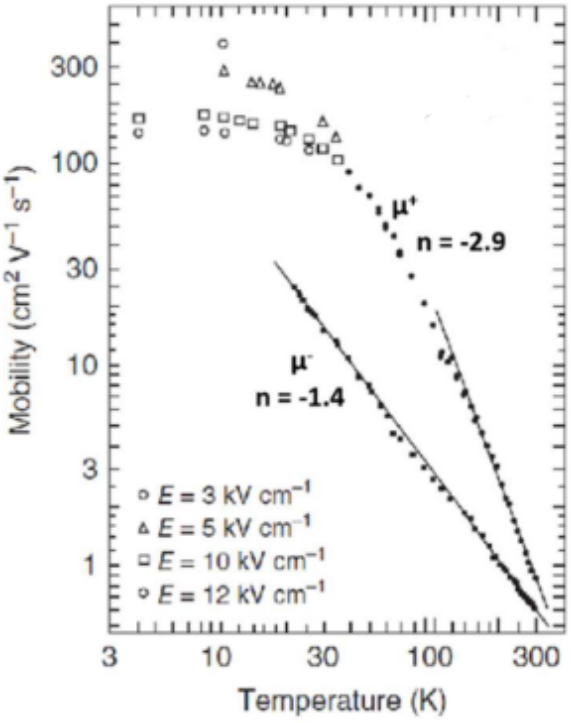


Figure 1.5: Mobility of holes and electron mobility in an organic single crystal against Temperature. The shape of the μ curve proves the band-like transport mechanism, such as that observed in organic ordered systems. Adapted from [174]

Electronic coupling

In particular, we have to introduce a new materials property called **Electronic Coupling**. This is a key quantity for the evaluation of charge transport in organic semiconductors. The electronic coupling in charge transport, in a nutshell, can be described as the interaction of the two Molecular Orbitals (MOs) where the electron occupancy is changed. More specifically, it the arises from the overlap of the electronic orbitals of the neighboring molecules, leading to the formation of hybrid orbitals with delocalized electrons. These delocalized electrons can then move freely between the molecules, enabling charge and energy transfer. This gives a strong indication of how easily

electrons can move between different states, and higher electronic couplings correspond to higher mobility and higher current flow.

1.1.2 Charge carrier traps in Organic Semiconductors

As it has been introduced in the previous section (section 1.1.1), charge transport in organic semiconductors is mainly based on the Van der Waals forces, which results in lower flow of current when compared to silicon-based semiconductors. Moreover, the reliance on these small forces also make them very susceptible to defect formation, and this can result in the formation of localized states in the band gap that can act as traps for the charge carriers[50]. Strongly influencing the transport mechanism, they can deeply alter the electrical and optoelectronic properties of the devices using these materials, significantly reducing their performance.
Based on the energetic distance between bands, we can distinguish two kind of traps:

- **Shallow traps**, closer to the HOMO/LUMO edge
- **Deep traps**, which are more distant from the HOMO/LUMO edge

Figure 1.6 visually shows this distinction. It is worth noting that shallow traps can be activated by thermal variations, and can play an important role in transport; deep traps, instead, do not suffer from thermal conditions and are usually a source of recombination.

In the next section, possible sources of traps will be introduced, together with a brief explanation of how they affect the transport in organic semiconductors.

Sources of traps in organic semiconductors

We can preliminarily divide sources of traps into two macro categories: intrinsic and extrinsic. Intrinsic sources are those that are independent from

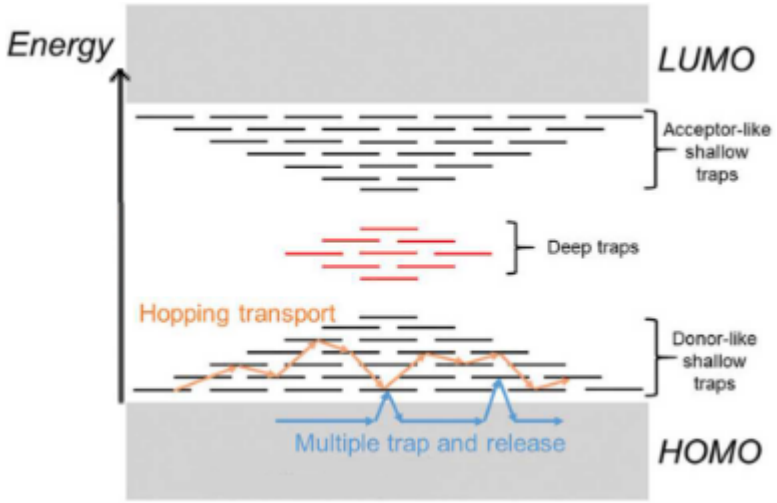


Figure 1.6: Spatial diagram of shallow and deep traps in organic semiconductors. Adapted from [50].

external factors, while extrinsic sources are those whose effect is influenced by elements of the context, external from the actual device.
Main sources are:

- **Disorder**: it can be of two types: dynamic and static. Dynamic disorder involves the entire molecular system and is the consequence of the thermal motions of the molecules, meaning that it is time dependent. Static disorder is instead caused by structural defects and chemical impurities (and is, consequently, time-independent) and its effects are local, affecting only the portion of the molecular aggregate where the defects are actually present. Structural defects are usually consequences of the deposition procedures (which will be introduced in the next section) and to the fabrication process in general. The spatial deformation induced by disorder induces the formations of localized trap states.

- **Interfacial Effects**: since OS devices are generally built by combining together many active layers, usually made with different materials, it

is important to consider the interactions that arise at the interface, i.e. where the organic materials come to contact with another material (being it organic or inorganic). These interfaces can be a source of traps due to non uniform local topology, energy variation, chemical interactions between the two materials, roughness of the materials surfaces and even the absorption of impurities like water, oxygen, etc. Particular attention should be paid to the interfaces between the semiconductor and the metal contacts; these can be the cause of the formation of many traps, affecting the injection or the collection of carriers and leading to high contact resistance[122].

- **Environmental Effects**: these can happen during fabrication, characterization and even when handling the device. Factors like temperature, environmental moisture, the presence of gases, radiation and humidity can all be sources of impurities and disorder, leading to the formation of traps[50].

1.1.3 Limitations

There are several challenges that need to be overcome in the development of organic materials in general and of organic semiconductors in particular. The main challenge is the intrinsic inefficiency of their charge transport mechanism when compared to the band transport typical of the silicon-based materials[2].
However, there are several parameters (like the degree of structural order, or the actual active molecular unit, and so on) that can be improved in order to obtain a better conducting material, leaving space for future optimizations and improvements. Particularly relevant are the potential effects of the introduction of dopant elements in the material, which can lead to a better flow of the electrons and to a more band-like transport.

[2]Silicon has a high electron mobility, which refers to the speed at which electrons can move through a material.

Another key challenge is the stability and lifetime of these materials, since organic materials tend to degrade over time due to factors such as the presence of moisture or oxygen, which can limit the lifetime of the devices that use them.

1.2 Electronic devices

With the term *electronic device* we intend a component used for controlling the flow of electrical currents for purposes like information processing, system control or energy production and storage.
There are many classes of electronic devices based on advanced and functional materials, each of which with its peculiarities, applications, functioning and active research lines. The common trait they share is that their quality depends on the materials used to build them, how they are fabricated and how the different components of the device interact with each other. Moreover, the environment in which they are built and used plays a fundamental role in their efficiency and longevity.
In this section introduce two generic classes of electronic devices based on traditional silicon-based materials, namely transistors and photovoltaic cells, showing their general features and working mechanisms. The description of traditional device architectures will serve as a general introduction for comparison with organic electronic electronic devices. The peculiar properties of the corresponding devices based on OS materials, which are at the basis of organic electronics, will be discussed in the description of specific applications (e.g., organic light-emitting diodes, organic field effect transistors, organic and hybrid solar cells) provided in the next chapters.

1.2.1 Transistors

Transistors are semiconductor devices that can be used to amplify or switch electronic signals. The most common type of transistor is the bipolar junction transistor (BJT), which consists of three layers of semiconductor

material: a base, a collector, and an emitter. When a small electrical current is applied to the gate, it modulates the electrons density (and, consequently, the flow of current[3]) between the collector and the emitter. This ability to control the flow of current with a small input current is the basis for the amplification and switching capabilities of transistors. Indeed, the main role of transistors is to be used as current amplifiers. When a small current is applied to the gate of a transistor, it allows a larger current to flow through the source and drain. The ratio of the drain current to the gate current is called the current gain of the transistor. The current gain can be used to amplify weak signals, such as those from a microphone or a radio receiver.
Transistors can also be used as switches: when the gate current is removed, the source-drain current stops flowing. This on-off switching action can be used to control the flow of electrical power to a load, such as a light bulb or a motor.
There are two main types of bipolar transistors: NPN and PNP. In an NPN transistor, the base and collector are made of n-type semiconductor material (i.e. a material which has a surplus of electrons in the outer energy levels), while the emitter is made of p-type material (which, in contrast, is a material with fewer electrons in the outer energy levels). In a PNP transistor, the base and collector are made of p-type material, and the emitter is made of n-type material. The direction of current flow through the transistor is reversed in PNP transistors compared to NPN transistors.

The Field Effect Transistor

While BJTs are the most basic and fundamental example of transistors used in the real world, in this thesis we focus more on field-effect transistors (FETs). The main difference between FETs and BJTs is that FETs do not use a base terminal to control the flow of current. Instead, they use an electric field to control the flow of charge carriers between the source and drain terminals. This makes FETs well suited for applications that require high

[3]It is important to note that this can be much larger than the one applied to the gate

input impedance and low noise.
Like a BJT, The FET is made up of three regions, which have different roles (and then names). These regions are called source, drain and gate. The source and drain terminals are made of n-type or p-type semiconductor material, depending on the type of FET. The gate terminal is separated from the source and drain by a thin insulating layer, such as silicon dioxide. The gate terminal is used to control the flow of charge carriers between the source and drain by applying a voltage to the gate-source terminals to create a potential barrier that modulates the flow of carriers.
There are two main types of FETs: the junction field-effect transistor (JFET) and the metal-oxide-semiconductor field-effect transistor (MOSFET). The JFET is the simplest type of FET and is made by reverse-biasing a pn-junction, which creates a depletion region that acts as the gate. The MOSFET is a more complex type of FET that uses a metal gate electrode instead of a depletion region. MOSFETs are widely used in digital and analog circuits because they have high input impedance, low noise, and low power consumption.
FETs are used in a wide variety of applications, including digital logic circuits, power electronics, and radio-frequency communication systems. They are also widely used as voltage-controlled resistors in analog circuits, where they are used to control the gain of amplifiers and the frequency response of filters.

1.2.2 Solar cells

A solar cell is an electronic device that directly converts sunlight into electricity. Light hitting the solar cell produces both a current and a voltage to generate electricity. This process requires a material in which the absorption of light moves an electron to a higher energy state; then this higher energy electron moves from the solar cell into an external circuit. The electron then dissipates its energy in the external circuit and returns to the solar cell.
The operation of a solar cell can then be summarized with the following

steps:

- Through exposition to light, the cell generates energy carriers
- The collected light-induced carries generate a current
- The current generates a voltage across the solar cell
- The power generated is dissipated into the external circuit

Architecture and operation of a solar cell

A classic solar cell is made of two layers, each made of a different type of silicon-based semiconductors:

- The **p-type**: this layer is made of silicon with added atoms of different elements. These elements are chosen between those that have one less electron in their outer energy level than silicon, like boron or gallium. This "missing" electron makes it impossible to create bonds with all the surrounding silicon atoms, giving birth to a so called "hole", i.e. a positively charged particle
- The **n-type**: on the opposite, this layer is made by adding atoms that have one more electron in the outer level than silicon (like phosphorus); this way, these atoms forms all the possible bonds with the adjacent silicon atoms, but one electron remains free from bonding, and is then free to move inside the silicon structure

A solar cell is then a n-type layer put on top of a p-layer. The point of contact between these two layers (and the closer portion of the two layers), is called p-n junction, which is the region where, when all the excess electrons of the n-type material and all the holes of the p-type material diffuse across the junction and recombine; when this process is completed, we see the formation of the so called depletion zone. Depletion zone is a space region that acts as a barrier for the flow of the charge carriers; because of the presence of these opposite charges, it creates an electric field that prevents the excess electrons

in the n-type layer from filling all the holes in the p-type layer[159]. Its width depends on the doping concentration of the semiconductors and the thickness of the junction. It is important to note that the width of the depletion zone also affects the efficiency of the solar cell, since a wider depletion zone will result in less recombination of the charge carriers, hence more current flow and more efficiency. When all the holes in the depletion zone are filled, the p-type part of the zone itself contains negative ions, and the n-type part contains positive ions instead.
When the solar cell is exposed to light, silicon electrons are ejected, leading to the formation of new holes. When this happens in the aforementioned electric field, this will move electrons to the n-type layer and holes to the p-type layer. If the two layers are connected with a metallic wire, the electrons will then travel from the n-type to the p-type layer, crossing the depletion zone and then going through the wire back to the n-type layer. This setting then creates a flow of electricity[159].

1.3 Materials and devices fabrication and experimental methods

As mentioned above, advanced materials, and in particular molecular and organic materials, can replace silicon-based systems for developing devices with new and improved functionalities. Moving from materials to full-scale devices, however, we need to take into consideration two main aspects:

- The interaction between two (or more) components of the device, usually thin-film layers made of two different materials. This gives birth to the so called interfaces; the phenomena that take place here are quite often very crucial in determining the properties of a device.

- The set of different processes and procedures used to fabricate the different materials and the techniques used to fabricate interfaces; these are as important as the materials used, since different ways to process

a molecular material can lead to completely different performances of the final byproduct

Moreover, the morphology of interfaces often plays a crucial role in affecting the performance of devices, making the fabrication process a fundamental factor in the final performance of a device. Here, we introduce the main concepts related to the fabrication and processing of molecular and organic materials and devices and a brief explanation of some of the main procedures used today.

1.3.1 Deposition Techniques

The process of applying a layer of organic material onto a pre-existing substrate is called *deposition* or *growth*.
Deposition can be done either with the material in the vapor or solution phase[4], and the choice is made depending on the specific vapor pressure or solubility of the material. Different deposition techniques result in materials with different morphologies and aggregation structures which, as discussed before, deeply affect the electrical properties of the semiconductor.
Even though there are several applications [5] of vapor deposition of semiconducting materials[31], at the time of this writing, solution techniques are more mature and reliable. Solution growth usually results in more uniform films of materials, and allows the realization of larger areas at relatively low temperatures. These properties result in scalable and flexible byproducts at low cost. The great progress made with these kinds of deposition techniques lead to a huge improvement of the performance of the resulting devices fabricated in the last years[78, 131, 130, 85, 127].
Here, we are going to introduce the main techniques used today, based on what is already present in many reviews[31, 176, 129, 86]. A summary is

[4]These means, respectively, the organic material opportunely vaporized or dissolved in an appropriate solvent

[5]Even for the development of flexible devices, which is an even more challenging field.

visible in figure 1.7.

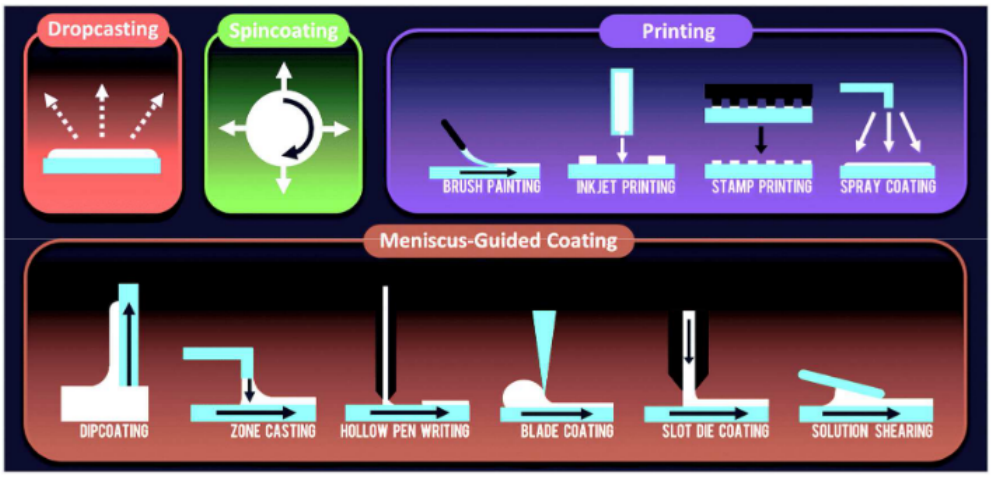


Figure 1.7: A schema of the main solution deposition techniques[31].

Drop casting

This is the simplest solution technique. It works by simply dropping the solution directly onto the substrate. The solvent is chosen in order to be able to spontaneously evaporate, leaving behind the organic materials in the form of a crystallized thin film.
The technique has been modified in recent years in order to optimize the crystallization of the semiconductor. One of these adaptations is Vibration-Assisted-Crystallization (VAC), where a delicate vibration is applied to the substrate. This way, the molecules gets enough energy to pass from its metastable state to a state of minimum potential energy, meaning the state with the highest degree of order[32]. Alternatively, Solvent-Assisted-Crystallization (SAC)[110] or specific surface treatments[45] makes the evaporation of the solvent slower, which is another way to improve the crystallization process.

Spin coating

This is probably the most used technique. The solution is deposited onto the substrate, which rotates at very high speed (usually more than 1000 rpm). The consequent centripetal acceleration forces the solution to spread uniformly on the entire substrate resulting, after the evaporation of the solvent, with a uniform semiconductor. The geometrical properties[6] of the resulting film are the consequence of many different parameters like spinning velocity and acceleration, the concentration of the solution and the type of solvent used.
Even this technique has been modified and improved over time. For example, in [181] it is presented a variation where the substrate is not placed on the rotational center, and this implies that the solution does not spread radially but along a specific direction.

Meniscus-guided techniques

Many deposition techniques based on solutions use some kind of linear translation of the substrate or the coating tool in order to allow for an aligned growth of the semiconductor. These methods usually rely on the formation of a meniscus on the solution, which in turn facilitates the evaporation of the solvent. Moreover, the relative linear motion of the solution and the substrate promotes a better alignment of the microstructure inside the organic materials.
Many parameters can be used to influence the crystallization process (for example the velocity of the translation or the temperature) and the selection of different tools result in many different techniques[40]. Some prominent examples are:

- **Dip-coating**: the substrate is immersed in the organic solution, then is pulled out with a controlled velocity. The evaporation rate and the velocity of the substrate are the main parameters, leading to different

[6] For example thickness, uniformity, microstructures and so on

thickness and crystalline structure of the final material[66].

- **Blade coating**: a "spreading element" (which can be a bar, a blade, a knife and so on) leaves a wet thin film on the top of the substrate, which is then left to crystallize. The velocity of the spreading and the temperature are still the main parameters that influence the quality of the final result. This is a very scalable technique, and has seen different adaptations[160, 135, 164]. In particular, a technique called Bar Assisted Meniscus Shearing (BAMS) has shown to lead to some of the best performing organic semiconductors based on small molecules.

Printing Techniques

These techniques give the possibility to deposit the organic material, solved in a solution, in a spatially confined manner, allowing the deposition process to follow a pattern.
Some important techniques are:

- **Inkjet printing**: it is probably the most famous technique of this kind. The process is made by spurting a droplet of the solution using a piezoelectric or a thermal process. The main parameter is the ink (i.e. solution) interaction with the substrate, which is in turn influenced by the surface energy of the substrate and the viscosity of the solution. An optimized version of inkjet printing is called Pneumatic Nozzle Printing[177], and is a combination of the inkjet printing approach with the meniscus guided one: the solution is "printed" by an outlet. This outlet is placed close to the surface in order to favor the formation of a meniscus. This method is perfect to obtain an organic film which shape follows a specific pattern

- **Spray coating**: this is another method based on an outlet sprouting small droplets. In this case, however, the droplets are aerosolized using an inherent gas as a carrier, then the particles hit the substrate and are able to dry very quickly, resulting in a very homogeneous film.

Main parameters are the pressure of the gas, the shape and dimension of the outlet, the concentration of the solution and the duration of the whole deposition process. This is another very scalable method that also allows to get very high quality materials spread on very large surfaces[82].

1.4 Computational methods

Nowadays, several decades have passed since computational science (and, in particular, computational simulations) has become a tool in the hands of researchers. By "computer simulation" we mean a process encompassing mathematical modeling of a scientific question where the modeling itself is done through specific software and digital tools, and the actual calculation is then performed on a computer. These softwares are meant and designed to predict the general behavior and the outcome of a real-world or physical system.
Computer simulations have altered the interplay between experiment and theory. The essence of the simulation is the use of the computer to model a physical system. Calculations implied by a mathematical model are carried out by the machine and the results are interpreted in terms of physical properties. Since computer simulation deals with models it may be classified as a theoretical method. On the other hand, physical quantities can (in a sense) be measured on a computer, justifying the term "computer experiment"[111]. While the reliability of these mathematical models could require validation by comparison with their results to the real-world outcomes they aim to predict, simulations are often easier and quicker to perform and organize than laboratorial experiments; even more, simulations can be performed in batches and concurrently, allowing researchers to test different hypotheses, materials and architectures all at the same time, in a identical environment and without the uncertainties of a real-world lab. Simulations also have another key advantage: they allow researchers to investigate and evaluate details and

properties of the physical entities that are otherwise impossible to analyze and know through laboratory experiments due to the nature of the phenomena, to the microscopic scale of the entities at hand or because of lack of reasonable ways to measure them.
However, these also have some drawbacks:

- In order to correctly and efficiently perform simulations, a researcher must be skilled in different complex disciplines and topics, and in particular:
  - Theoretical physics and chemistry
  - Applied physics and chemistry
  - Specific approximations and assumptions of specific algorithms and methods
  - Computer science and engineering
  - Programming
  - High-performance computing (HPC) environments
- While being quicker than actual experiments, simulations are still a long and complex computation, which can last from minutes, to hours or days and even weeks or months
- Large scale simulations require HPC facilities
- Many of the methods are intrinsically based on numerical algorithms, for which the accuracy and convergence must be tested and validated before, during and after the computation

Here, we are going to introduce some of the main computational methods used at this date.

1.4.1 Density Functional Theory

Density Functional Theory (DFT) is a computational quantum mechanical modeling method used in physics, chemistry and materials science to

investigate the electronic and nuclear structure of many-body systems, in particular atoms, molecules, and the condensed phases. Using this theory, the properties of a many-electron system can be determined by using functionals, i.e. functions of another function. In the case of DFT, these are functionals of the spatially dependent electron density. DFT is among the most popular and versatile methods available in condensed-matter physics, computational physics, and computational chemistry.
DFT is an example of ab-initio method, meaning that it require no information coming from empirical knowledge about a physical/chemical system analyzed; instead, it uses many different approximations to solve the Schrödinger equation using wave functions for the description of the atomic orbitals and the calculation of molecular properties[12].
The history of DFT is rooted in the birth of quantum mechanics in the early 20th century. Applying quantum mechanics principles to more complicated systems such as molecules and solid-state materials proved to be difficult: even in classical physics there is no general solution to a three-body problem[7], but in order to describe just a water molecule at the quantum level we have to deal with ten electrons and three atomic nuclei.
In this context, DFT emerged in the mid-60s from a single idea: tackling this problem by not focusing on the individual electrons but instead using the electron density as the fundamental variable to solve for, and furthermore reformulating the many-body problem as an equivalent single-particle problem.
This single idea help to solve two categories of problems:

- Handle any element in the periodic table in any kind of atomic arrangement, without the need for experimental input parameters. Because of this, DFT has strong predictive power, even for completely new molecules or materials. That made atomistic simulations able to reduce development time and cost; using HPC clusters, a single researcher can screen hundreds or even thousands of materials in parallel, vastly out-

[7]For example: the combined orbital motion of the sun, the moon, and the Earth

numbering the number of experiments a human can perform at the same time

- Understand how materials and devices behave and operate under different conditions. A trained DFT user can correlate measurement data with simulation results to draw conclusions about the physical origin of certain effects observed in the material or device that cannot be explained with other, simpler models. Such insight is crucial in order to fully understand the properties of the analyzed material, to scale down device dimensions or optimize materials choices or process conditions

1.4.2 Molecular Dynamics

We can describe molecular dynamics (MD) simulation as a technique where the atomic trajectories of a system made of N particles are computed by numerical integration of Newton equation of motion starting from certain initial and boundary conditions. The main idea behind MD is that of computing the property of the system at hand as a temporal mean, based on the trajectories of the N particles present in the system. This is done by enforcing all the methods known as statistical mechanics[111].
In a nutshell, the main idea is that, known the potential, the force acting on the ith particle is given by the gradient with respect to the atomic coordinates; so, starting from the Newton equation of force:

$$F_i = m_i \frac{d^2 r_i(t)}{dt^2}$$

Assuming that m_i is the mass of the ith atom, $r_i(t)$ is the position at time t, then F_i is the force acting on the particle i at time t.
With regard to the force related to the potential, the equation is:

$$F_i = -\nabla U(r_1, \ldots, r_N) = -\left(\frac{\partial U}{\partial x_i}, \frac{\partial U}{\partial y_i}, \frac{\partial U}{\partial z_i}\right) \text{[111]}$$

As such, MD can simulate the dynamical behavior of complex systems in given thermodynamical conditions. In several application contexts in the

field of materials development, MD simulations are used in conjunction with molecular mechanics (MM) potentials. Here, a molecular system is generally described in terms of a set of very simple potential energy terms, approximating the constituting units as a set of particles kept together by mechanical interactions. Although very rough, these approximations allows the simulation of very large molecular aggregates at relatively limited computational costs. The drawback of this approach, however, is the intrinsic inability to describe systems where chemical bonds are formed or broken, thus at difference with electronic structure methods (e.g., DFT). A full example of MM potentials could be:

$$\begin{aligned} U\left(r_1, r_2, \ldots, r_N\right) = & \sum_{i_{bond}=1}^{N_{bond}} U_{bond}\left(i_{bond}, r_a, r_b\right) \\ & + \sum_{i_{angle}=1}^{N_{angle}} U_{angle}\left(i_{angle}, r_a, r_b, r_c\right) \\ & + \sum_{i_{dihed}=1}^{N_{dihed}} U_{dihed}\left(i_{dihed}, r_a, r_b, r_c, r_d\right) \\ & + \sum_{i=1}^{N-1} \sum_{j>i}^{N} U_{pair}\left(i, j, |r_i - r_j|\right) \\ & \ldots \end{aligned}$$

Here, we can introduce the different components of these potentials, that are:

- **Bonds**: these are made of two particles (atoms) which are connected by a strong link, mimicking the occurrence of chemical bonds through electron sharing
- **Angles**: these are made of three particles, linked by a bond in a two-by-two fashion
- **Dihedrals**: like the angles, but made of four particles instead of three

- **Pairs**: similar to bonds, but instead of sharing electrons, their link is due to other interatomic (non-bonding) forces.

The definition of full potential terms enables MD simulation of complex systems. An ordinary run of a MD simulation can be represented as in Figure 1.8:

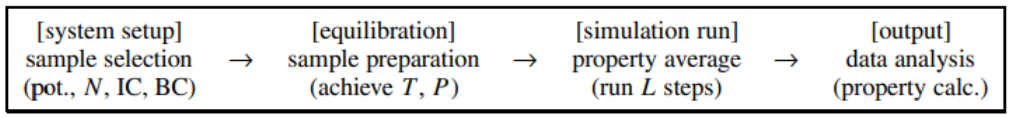


Figure 1.8: a schematic of the general MD simulation workflow

MD simulations have to deal with a complex problem: the selection of the atomic force field, i.e. the set of interatomic forces that bind each pair of atoms/particles present in the system. In particular, the potential U represents the potential energy of N interacting atoms as a function of their positions. These potentials can be built in two different ways:

- Using empirically determined values for many possible atom pairs (i.e. an hydrogen atom bonded with a carbon atom) but also depending on the chemical neighborhood
- Computed using more accurate lower level methods (like DFT).

In the first case, it is easy to understand how building the potential for a system with many atoms of different atoms types using this method can be very difficult, even more so because not all the possible pairs and neighborhoods have an empirically determined potential. This leaves the researcher with the burden of manually compiling the potential for the system at hand, carefully choosing the different potential types for each o. However, due to the complexity of the task and to the incompleteness of the known, empirical potentials, part of this process is inevitably led in a trial-and-error fashion, and is more a work of art and experience than a precise science. In the second

case, researchers have to first develop DFT simulations, perform them and then use the results to perform the subsequent MD simulation, meaning that two different simulations enviroments (and so two different set-ups, software stack, configuration options and so on) must be taken into account.
Another key limitation of MD simulations is related to simulation time scales. That means that a lot of interesting phenomena (even for applications like biology, chemistry and so on) take place both in very restricted time frames and very long ones (i.e. time intervals ranging from femtoseconds to seconds); this means that a simulation should last for billions of time steps to capture all these different phenomena. Since this kind of simulation is usually performed for very big systems (in the order of magnitude of tens of thousands of atoms) these kind of computations may result to be intractable. This is also related to the rare events problem; here, "rare" is relative to the amount of events that happens at this scale, so these rare events (phase transitions, transformations of the elements and so on) actually happen many times for each microscopic interval of time. But computers are only able to compute all the equations at discrete time-steps, and for these reasons we have a high probability of missing many of these rare events. Unfortunately, these events are also the most fundamental ones for deeply understanding the physics and chemistry of a system, and the inability to capture them with MD leads to imprecise results that do not take into account the impact and effect of these events.

1.4.3 Device simulations

These are higher scale simulations, which are grounded in statistical models of all the different kinds of phenomena that determine the functionality of real-world devices. Differently from the two previous families of methods, the main difference is that device simulations method lose any kind of description of the atomic nature of the materials and phenomenon analyzed, and the physical system is described using a (1D/2D/3D) mesh for the discretization of the system. On these meshes, the equation describing the system or

the phenomena investigated are then computed. These equations are generally differential equation based on empirical knowledge, results coming from lower scale simulations or from theoretical frameworks used to describe the system.
These simulations can be of many different types. Here, we are going to highlight the main ones.

Drift-diffusion simulations

Drift-diffusion equations are a system of partial derivative equations used to describe the charge transport phenomenon of electronic devices. They have a particularly important role in the simulation of properties of devices based on semiconductors[154, 107].
The equations are:

$$\begin{cases} -\nabla \cdot (\epsilon \nabla \varphi) - \rho = 0, & in\ \Omega, \\ q\dfrac{\partial n}{\partial t} - \nabla \cdot \mathbf{J}_n = q\left(G - R\right), & in\ \Omega_{semic}, \\ q\dfrac{\partial p}{\partial t} - \nabla \cdot \mathbf{J}_p = q\left(G - R\right), & in\ \Omega_{semic}, \end{cases}$$

where:

- Ω and Ω_{semic} are the region of space occupied by the device and that occupied by the semicondutor alone respectively
- ϵ is the electric permittivity of the material
- q is the element charge
- φ is the value of the electrical potential inside the device
- n and p are the density of the two charge carriers, namely electrons and holes respectively
- ρ is the charge density per volume unit; in particular, for the semiconductor holds: $\rho = -q\left(n - p + D\right)$, where D is the concentration of the eventual doping elements; for the insulator, this is simply $\rho = 0$

- $\mathbf{J}_n$ and $\mathbf{J}_p$ are the density vectors for the electric current, normalized per surface unit
- $(G - R)$ is the generation-ricombination ratio

This must be paired with careful defined boundary conditions[8] and then solved with regard to φ, n and p as functions of time and space.

Multiscale simulations

The aforementioned methods have different strengths and drawbacks, but share a trait: limited by their weaknesses, they are unable to simulate all the properties needed to fully understand an actual physical system, ranging from quantistic properties to the actual physics of a full device together with the properties of the materials that compose the device.
In order to overcome this problem, the modern approach to simulation revolves around the joint usage of many of these solutions, using information and results coming from one or more different scales to aid the convergence of a specific method or to obtain more precise results. This way of thinking and approaching the problem is called *multiscale simulation*.
The different scales and their order are depicted in figure 1.9.

While being a more powerful approach, this also gives birth to new problems, mainly due to the increased complexity of the approach. In particular, merging the information coming from different scales is not a trivial task, both due to the different physical model used at different scales and, consequently, the different softwares used to perform the different simulations. These softwares also use very different data structures and file formats, furtherly impairing the integration of the respective results. This process is even more difficult when the problem analyzed is so complex that it requires a mixture of a bottom-up[9] and top-down[10] approach, leading to the need of

[8] These are dependent on the geometry of the device

[9] Meaning that we are going from a lower scale (for example, quantum physics) to a higher one (for example, molecular dynamics)

[10] The opposite of the bottom-up approach; for example, we may need to use results

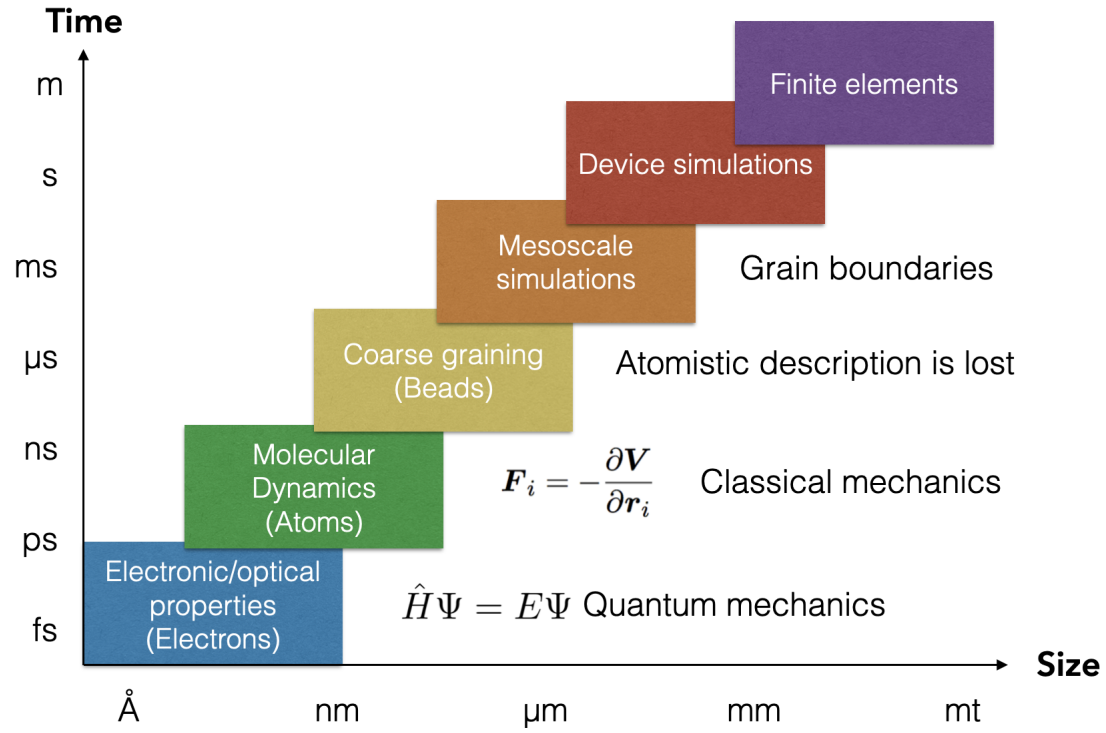


Figure 1.9: A figure showing the different scales of simulations in a plot, showing their distribution with regard to the size scale (on the x-axis) and time (on the y-axis)

integrating different results coming from different scales more than once and, possibly, in both directions.

This is a situation where automatic and data-driven techniques can play a fundamental role in helping researchers to improve this mechanism, both in terms of ease of use and performance-boosting. Moreover, data-driven techniques can also help to make the different scales more integrated, and there are some pieces of literature already moving in this direction[63].

stemming from a device simulation into a coarse-grain simulation

1.5 Data, an open challenge in applying Data-Driven techniques to Materials Science

Here, we discuss the general picture of the state-of-the-art of data-driven techniques applied to materials science, presenting the main problems at a high level. Then, in each chapter, we will analyze the deeper technical aspects of these problems, together with the proposed solutions.
At the time being, the field suffers from a deep fragmentation of software, techniques and file formats, often customized for a specific research field (protein dynamics, pharmacology, single-molecule analysis, device optimization and so on). This means that sharing information and data coming from different sources is very difficult because they are usually encoded in different formats, which sometimes may not contain all the data needed to be used in a different context with a different software for a new purpose. This means that researchers have to invest a lot of time in converting files or even gathering missing information from papers, other files or contacting the original author in order to be able to use data for their work, and this is a huge problem both for new research lines or for reproducible science.
Another problem of the actual state-of-the-art in computational simulation is the need to bend the actual physics of the problem at hand to the needed approximation, and that sometimes means manually filling long and complicated input files (which, again, are custom-formatted in order to be used with the specific software used). For example, in order to perform a MD simulation, a researcher must write the entire force field for all the entities at play; this can be done by consulting very long and specific look-up tables reporting the best parameters for a specific pair of atoms. However, these values do not depend only on the two atoms that form the pair, but also on the larger chemical neighborhood surrounding those atoms. This means that it's not uncommon to have a combination for which the exact force field is not known, and this leaves the researcher with the burden to identify the best approximation between those already known. This is a very difficult

task based on very specific expertise, resulting in a high entry barrier to the discipline for young or new researchers.
The whole process is even harder when a scientific question requires to be tackled on different scales. In fact, while the approach is relatively similar, the kind of input files that must be prepared, the resources that have to be used and the physical and technical knowledge required are very different, leading a lot of researchers to specialize in one specific scale. This means that leading complex simulations, investigating a problem at all the possible scales, is usually a job for very large, multidisciplinary teams. Moreover, each step can be completely disjoint from the next and previous one, and integrating them may require an additional process of tuning different simulation parameters in order to make the whole pipeline consistent.
Another important aspect to consider is the fact that each of these simulation steps can be performed with a plethora of different software, any of which uses its own standard for input files, naming conventions, specific simulation parameters and, obviously, output file formats and information. In the seminal years of the discipline, researchers have been pushed to develop their in-house software and corresponding data formats, in a quest for developing the most powerful simulation stack, while embracing the new technologies and languages developed along the years. This effort led to the development of many, very efficient softwares like CP2K[83], GROMACS[15] and LAMMPS[166].
However, this led to a situation where data formats for results, input files and computational recipes are completely customized to each software; even the files for describing the actual chemical entities (like molecules, sets of atoms etc) can be encoded in one of many different formats, where some are completely barebones (like *.xyz* files, which actually contains only the coordinates and types of the different atoms) or very complex (like *.pdb*, the format developed and used for the Protein Data Bank[16], where many advanced information and metadata are stored into the file). It is easy to understand how this situation makes it very hard to design new experiments while also

enforcing the knowledge and results obtained by other teams and researchers who are using different software and file formats. Also, as we are going to discuss more deeply in other chapters who are fully dedicated to machine learning (namely chapters 2, 4 and 5), this scattered data is very hard to be put to use for training machine learning models; in fact, collecting data coming from different experiments mean dealing with the aforementioned plethora of formats, files and encodings, making it long and complicated for the data scientist to be able to integrate all the different data sources in other to gather sufficient amount of data.

Some work has been made in order to overcome these limitations[3], but these solutions are only partial; for example, software like Galaxy[3][11] are meant to give researchers a way to automatically execute their simulation workflows using small pre-built building blocks, while automatically collecting the produced data with the specific pipeline that produced them, saving also an history of potential multiple runs. It is obviously a very powerful software and promising approach, but it also has some drawbacks; the main idea behind Galaxy is to wrap every component inside an XML[138] scheme, allowing the software to track everything and maintain a record of provenance and correspondence in a univocal way. However, this adds a new layer of complexity to an already very nested, stratified and complex discipline, which means that researchers who cannot find an implementation of one or more of the building blocks of their experiments then need to learn both how to implement such a building block and how to "wrap" it inside an XML file. Moreover, it looks like not all the existing simulation software is available by default on Galaxy, which means that researchers need either to learn how to use a new simulation software or must find a way to integrate the one they prefer inside Galaxy. Ideally, we would like to give researchers tools like Galaxy but without this added complexity.

[11] While Galaxy has been initially developed for biomedical applications, it has already been used also for other domains, and many of the software embedded inside Galaxy (like Gromacs) are the same software used in materials science that we introduced before.

Chapter 2

Developing features for materials entities

In this chapter, we introduce the work done on applying machine learning algorithms to the prediction of properties of the simplest constituting units in aggregates of molecular materials, which are molecular pairs. It must be noted that there is a radical difference between dealing with single molecules or dealing with molecular systems or molecular aggregates (even the small ones). These differences are discussed throughout this first chapter and the related activities, but they are going to be analyzed in further detail in the following (i.e. chapter 3).

At the beginning, we give a deeper analysis on the state-of-the-art of machine learning applications to the case of molecular systems, highlighting strengths and limitations. Then, we focus on the development of features that are able to describe the physics of the systems at hand, allowing for more powerful predictions even in more complex cases. Moreover, we focused on creating models that are trainable even with small datasets[1], which is a common

[1] In this project, we are working with datasets coming from actual simulation workflows. These simulations tend to be very long and difficult to set-up, which usually means that the resulting datasets are quite small. Improving the computational throughput of this kind of workflow is, as already stated, one of the incentives of using machine learning techniques in this realm.

scenario in the materials modeling field due to the expensiveness and duration of traditional simulations. It must be noted that this is the lowest possible scale of simulation used in multiscale simulation approaches, which is another reason why we decided to tackle this problem at the beginning of our research activities.

2.1 Physics and chemistry context

In this chapter, we are going to use the knowledge introduced in the previous chapter, and in particular in section 1.1. In particular, we are going to deal with the **Electronic Coupling**, as introduced in section 1.1.1. The computational evaluation of charge transport properties in molecular materials generally requires the simulation of electronic coupling values within a large set of neighboring molecules constituting a model of a molecular aggregate. It is then easy to understand why it's crucial to develop methods to compute the electronic coupling that are both reliable and efficient in order to enable the possibility to quickly use the computed couplings to then compute higher scale properties of bigger aggregates.

2.1.1 A computational perspective

Generally speaking, the properties of molecular materials depend both on the properties of individual molecules (related to the chemical composition and structure of a molecule) and on the properties of aggregates. For example, in Fig. 2.1 the complex structure of the morphology obtained from molecular dynamics simulations of aggregation of a perylene diimide derivative is shown[100, 98].

The aggregation morphology depends, in turn, on both the peculiar molecular structure and on processing conditions and environment[101]. In several cases of technological interest, the resulting aggregate exhibits structural

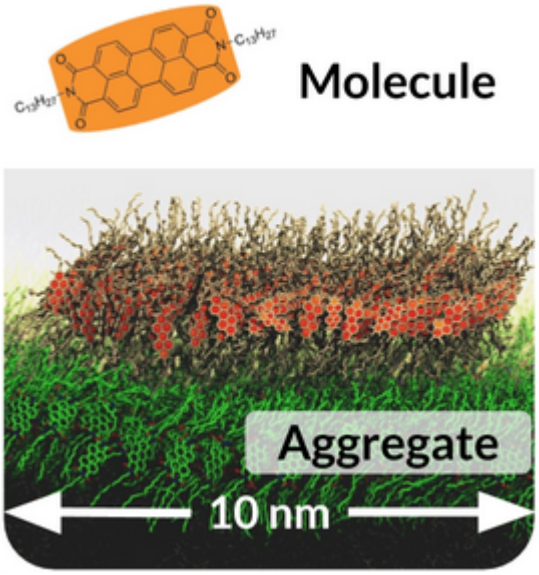


Figure 2.1: Molecular structure of a perylene diimide derivative and resulting simulated morphology at the interface with a substrate.

features on the nanometric scale. Indeed, nanoscale aggregation and morphology have impact on several properties of molecular materials[152]. The evaluation and prediction of the properties of aggregate must therefore consider the properties of materials across a quite wide range of length scales, from the molecular scale to the nano- and micro-scale.

Multiscale simulations techniques provide tools for the modelling of the properties of materials at different scales[170, 34]. In the particular cases considered in this work, multiscale simulations can be used to link the properties of individual molecules to the properties of molecular aggregates. Specifically, different computational methods target phenomena occurring at different scales, and the output of a simulation at a given scale can be used as an input to perform another set of simulations at a lower or higher scale, providing the cross-scale link.

A particularly interesting case study concerns the evaluation and prediction of the charge transport properties in molecular semiconductors. The

charge transport properties of molecular materials are exploited in several cases of technological interest, for example in the development of organic light-emitting diodes (OLEDs) or organic photovoltaic (OPV) solar cells. In several cases, the propensity to efficient charge transport depends on the intrinsic electronic properties of materials, as for example occurring in functionalized carbon-based nanostructures[115, 155, 113, 114, 112]. In the case of molecular materials, however, the overall properties of the materials, in terms of phenomena related to charge transport, depend on:

- the electronic properties of individual molecules (electronic configuration, energy levels, etc.);
- molecular aggregation, intermolecular interactions, morphology, deformations, interfaces and all other effects concerning the interaction of individual molecules with other molecules or materials[97].

A very simple, though effective, model of molecular semiconductors describes the charge transport process in terms of percolation of charge by hopping from a molecule towards a neighbouring molecule, as shown in Fig. 2.2.

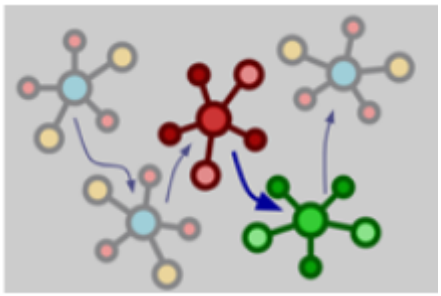


Figure 2.2: Charge hopping mechanism in molecular semiconductors.

The current flowing through materials can essentially be interpreted as a sequence of single events involving two neighboring molecules. This quantity can usually be determined by DFT simulations, involving pairs of molecules and neglecting collective effects. The transport properties of molecular ag-

gregates can subsequently be obtained by a sort of weighted statistical integration, for example by applying kinetic Monte Carlo (kMC) simulations[134, 168]. However, two relevant issues must be considered:

- The evaluation of the intermolecular couplings by DFT simulations is quite demanding, from the computational side, and can require up to a few CPU hours for a single molecular pair, on standard computational infrastructures.
- As we discussed before, the properties of molecular materials, including charge transport, depend strongly on the aggregation morphology and on resulting interactions on a scale of several tens or hundreds of nanometers. As the typical size of individual molecules is on the order of a few nanometers, the evaluation of intermolecular coupling in nanoscale aggregates results in several thousands of pairs, each of which needs an independent DFT calculation.

The use of statistical methods, such as Kinetic MonteCarlo (kMC), for the evaluation of charge transport properties requires a balance between accuracy and computational load, which can exceed several thousands of CPU hours. Therefore, we need a set of tools which can assist the evaluation of the charge transport properties of molecular materials with good accuracy and, possibly, saving CPU time.

2.2 Machine learning for materials entities: state of the art

In recent years, machine learning (ML) methods have applied with success to studies of the properties of molecular materials[125, 77, 58, 21, 150, 120, 149]. The vast majority of these studies are focused on the properties of individual molecules, targeting the correlation between molecular structure and resulting properties[53, 93, 147]. The properties of several technological materials constituted by molecular aggregates, however, depend on both

molecular structure and on aggregation morphology, as for example in the case of nanoscale materials[91, 90]. Computational methods for predicting the properties of molecular materials must therefore integrate the properties of individual molecules with information about aggregation morphology, which, in turn, can be related to materials fabrication and processing[97]. The definition of a modeling paradigm able to simulate and predict the properties of molecular materials as a function of molecular structure and aggregation/fabrication conditions can potentially enable high-throughput development of novel materials for technological applications.

2.3 Development of a systematic approach

After the analysis of the current state-of-the-art and the limitations discussed in the previous section, here we introduce our approach to the application of ML to materials science. In general, we can identify two macro-approaches present in literature:

- Using physics-based featurization of the entities at hand, which allow us to use shallow and simple models to fit the problem (we can relate this to higher scale simulations)[36, 183]

- Using powerful model with non-specific features which can capture some aspects of the physical entities at hand (we can see this technique as more related to ab-initio methods)[133, 52]

The first approach suffers from a common problem: they lack a systematic approach to the application of machine learning algorithms to materials science and molecular entities in general, leading to sparse results and ad-hoc procedures with little potential for re-application to different domains and tasks or even to the same task applied to different molecules. Moreover, many of the procedures developed in these works are only usable with simpler entities (small molecules, linear molecules, restricted number of atoms)

and do not study the impact of different ways of representing the same characteristics.
On the other hand, the second class of articles enforces very powerful features and data format like graph representation of molecules, which suffers from high-complexity both in the conversion of raw structural files to these high level features and in their actual usage both from a computational load standpoint and with regards to the models that are able to process them for their learning phase; in particular, they require longer training times, more data entries and customly developed models (i.e. graph neural networks[141, 24, 175]) which despite being very powerful and promising are also hard to train and require an amount of data that is hard to have at disposal in the materials science domain[184, 23, 151, 180, 9].
In this work, we are introducing a first attempt at systematically developing lightweight yet expressive features to represent molecular systems that are able to be fed to rather simple models while also being able to be used to solve related tasks for different (and even very different) molecules with no modification required. A specific work on featurization is required for different reasons:

- from a technical point of view, the amount of data available to researchers if often too limited to allow for reliable and fast enforcement of deep learning techniques

- having simpler, more interpretable models is an important factor when the problem at hand has no prior known answer and the AI approach is used in order to help researchers find the root principle of a chemico-physical effect instead of just finding the answer for a specific molecule or material in a specific context

- on the opposite, if a proved rule is known, enforcing these rule to extract meaningful features can serve both as a way to reduce the computational burden of training an algorithm and as a method to further investigate the deeper nature of a known phenomena

Obviously, complex models play a fundamental role in scientific research, with very notable examples like AlphaFold[72], but they serve a different purpose, that is giving answers for unknown problems helping researchers formulate new hypotheses and finding new rules or laws previously unknown[2]. However, we think that after this step researchers could furtherly confirm their new findings by creating (or, hopefully, reusing) specifically tailored featurization of the entities at hand to train a simpler model: if this model achieve comparable results to the more complex ones, then this could be another good sign of the validity of the new theoretical[3] findings.
Ultimately, our target is to be able to describe intrinsically complex multi-scale systems relying on simple techniques, enforcing knowledge about the physical systems and materials analyzed to optimize both the computational load and the shareability of the implementation. In this specific case, we want to find and fit the correlation that exists between the structure and the morphology of a given material (and, in particular, a molecular pair) and a target property.

2.3.1 A multiscale top-down approach: from simulations to data workflows

Our approach relies on a top-down view of the properties of molecular materials for applications. For example, we can consider the properties of active materials used in organic electronic devices as derived from interlinked materials properties on progressively lower length scales, from the device to the molecular scale, as shown in Fig. 2.3.

In this case, we can first consider the aggregation morphology of molecular

[2]Another important role of complex models is for those problems where a proven answer does exists, but the chaotic nature of the physical process at hand makes hard to rigorously and reliably calculate it for specific contexts with traditional (i.e. procedural and/or non-statistical) computational tools

[3]And, as and added bonus, this theoretical findings are immediately usable in practical context via the trained models

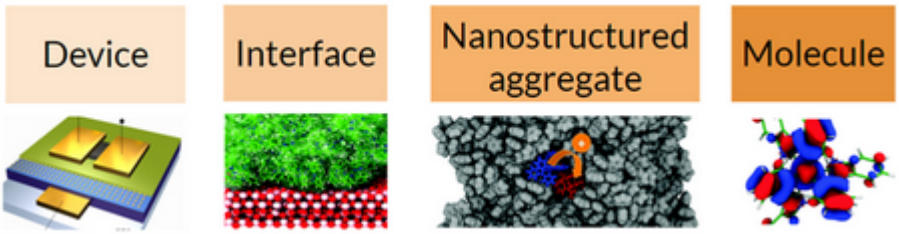


Figure 2.3: Top-down description of the properties of active materials used in organic electronic devices. Partially adapted from Ref. [79, 10, 97].

materials at the nanoscale. For example, we can simulate the aggregation of molecules, in different conditions, by atomistic (or coarse-grained) MD[96]. This step will also link the nanoscale morphology of molecular materials to processing or fabrication conditions, a fundamental part in the engineering of organic electronic devices[99]. Then, we can proceed to a reduction of the scale, extracting pairs of neighboring molecules from the MD configurations and computing electronic couplings for each pair (see Fig. 2.4). As explained before, however, this step may require the evaluation of electronic coupling for a large number of molecular pairs, in the order of thousands or more.

It is worth noting that the top-down approach discussed above relies, technically, on the knowledge of the molecular structure only. Indeed, the aggregation morphology of molecular materials, at least for pure bulk materials, depends on the molecular structure and aggregation conditions only. The whole process that goes from the single molecule to the final pairs selection can therefore be represented as in Fig. 2.5.

We start from the knowledge of the structure of the individual molecule. On the basis of this knowledge, we build a suitable atomistic potential, usually in terms of a force field, including intramolecular and intermolecular terms. We select the conditions leading to aggregation and build a computational model that is able to reproduce the aggregation morphology using MD simulations. The individual molecular pairs are extracted from the simulated aggregate, and DFT calculations are carried out for each selected pair. This

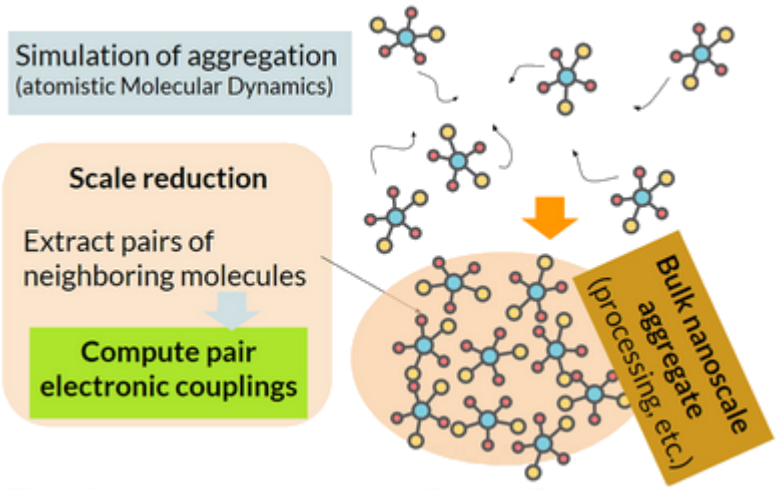


Figure 2.4: Simulation of the aggregation morphology of molecular materials by MD, from which individual pairs are extracted for subsequent DFT calculations.

set of steps also defines a flow of data which links molecular structure to charge transport properties.

2.3.2 Selected molecules and task

The general task of this activity is to develop a general set of rules to create machine learning-powered software for predicting the properties of molecular aggregates. These properties are determined by the structure of the molecule(s) that compose them and by the way they arrange themself with regard to the others in the bulk. For these reasons, we aimed at developing a set of features and procedures able to use this kind of spatial information and relate them to the properties at hand.
In order to be able to come up with the most general solution possible, we chose a very complex molecule as our testbed and we started from the simplest aggregate possible: a molecular pair. In particular, we decided to work with the fac-tris(1,3-diphenyl-benzimidazolin-2-ylidene-C,C2')iridium(III)[81] (also known as DPBIC), a molecule that has applications in organic electron-

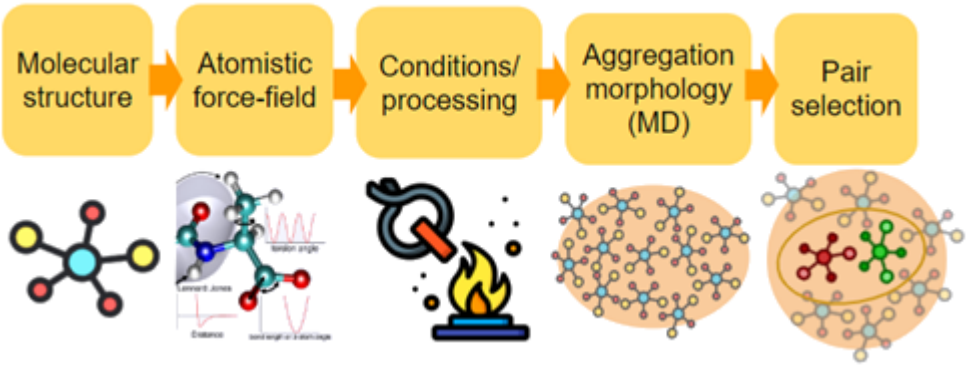


Figure 2.5: Multiscale workflow for the simulation of charge transport properties in molecular aggregates.

ics and in particular in OLEDs systems. We chose this molecule because of its geometrical and topological characteristics:

- It has a considerable amount of atoms (namely 103), making it a fairly big molecule
- It has a pseudo spherical shape and symmetry
- It is used as the basic component for an homonym molecular material, whose properties strongly depend on how the single molecules arrange themselves in space absolutely and respectively

Our aim is to define a set of features that gives us the possibility to train a model in order to determine the properties of the basic building block of a bulk of such material, a pair of molecules; as a results, having the ability to predict the selected properties for a molecular pair will then give us the possibility to enforce well established techniques (i.e. Monte Carlo methods) to calculate the aggregated property of a bigger bulk.
In our example, we selected the diabatic electronic coupling, which is an important property for materials used in (organic) electronics in order to achieve good performance (high luminous efficiency, power efficiency and so

on). This property is known to depend, aside from the actual molecule or molecules at play, on the distance between the molecules and on their mutual respective orientations. More details on the physical problem are given in the next subsections.
Other than being good for training the machine learning algorithm, we want to develop a set of features that are also understandable for human researchers, giving them the opportunity to understand both the physical characteristics of a system and to interpret why, once trained, the model actually gives specific predictions for a specific case; this way, researchers are able to both debug a non-working model (for example: a model that gives unreasonable predictions for a well known situation) or to get insight on cases for which they had no prior knowledge and results.

Standard computational workflow

Our specific problem can be described as follows: we can use density functional theory (DFT) calculations to compute the electronic coupling between dimers extracted from molecular aggregates. Essentially, the intermolecular electronic coupling represents the propensity of charges to jump from one molecule to a neighboring molecule in a given molecular pair. The knowledge of the electronic coupling in all possible molecular pairs in a given set of molecules allows for example the simulation of the electrical current passing through the molecular aggregate.

The simulation of electronic couplings in molecular aggregates proceeds as follows:

- The morphology of a bulk amorphous aggregate of a given molecular system is simulated by molecular dynamics (MD). In this step, individual molecules are inserted into a periodic simulation box and a suitable MD protocol is applied to induce aggregation until a target density is reached. A high-symmetry (e.g. cubic) periodic box can be used, as periodicity should not impact (for large boxes) on the morphology of the aggregation. A box size of about 10x10x10 nm is generally sufficient

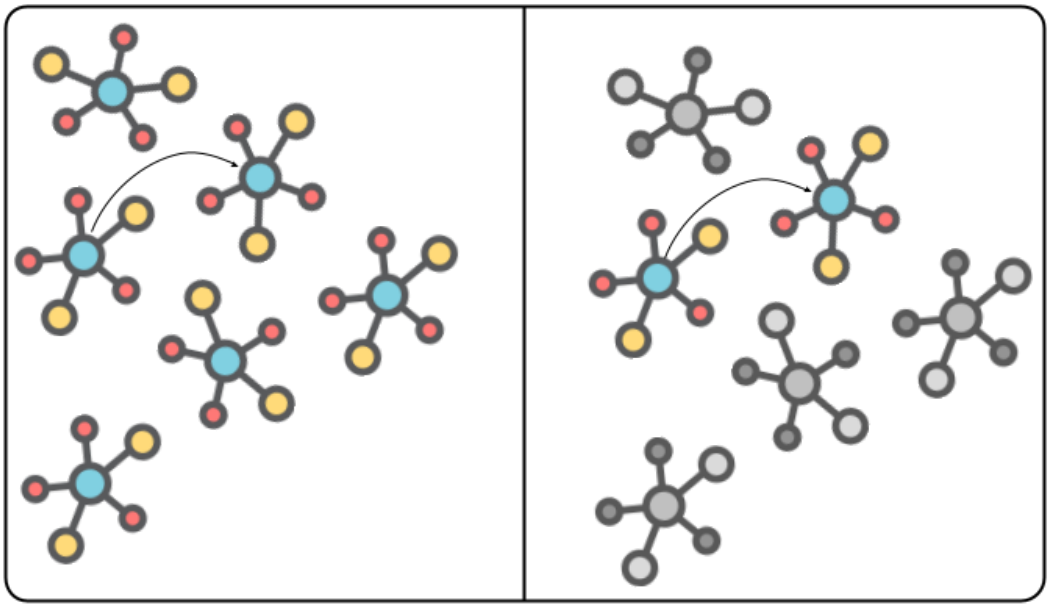


Figure 2.6: Pairs of nearest-neighboring molecules are considered from the morphology of a molecular aggregate (left). The selected pair is isolated from the aggregate and the electronic coupling between the two molecules is computed.

for the simulation of currents in amorphous aggregates. This leads to several hundreds of molecules in the simulation box (with a molecular density of about 1 molecule per nm^3). The system is then equilibrated using MD, and a configuration file is obtained with the position of all atoms in all molecules in one of the equilibrated configurations. If we have N molecules with M atoms in each molecule, we will have a total of NxM atom positions, consisting of a record of 3 real coordinates in 3D space (x, y, z).

- Pairs of nearest-neighboring molecules (that is: molecules with a distance between the respective centers of mass below a given threshold) are extracted from the aggregate. Either a set of random pairs or the whole set of pairs in the periodic box can be selected from the config-

uration file of the equilibrated aggregate.

- For each selected pair, the configuration of the two molecules (say, A and B), in terms of 3D coordinates of all constituting atoms, and atom types (atomic number) are considered. These coordinates are used in a DFT calculation of the electronic coupling between the two selected molecules. This step is repeated for each selected pair of molecules.

At the end of this computational workflow, we will have a correlation between atom pairs and electronic coupling, for example as a look-up table, connected to the coordinates of each molecule.
It is easy to see that this is quite expensive (a few CPU hours for each pair) and must be repeated for a large (several thousands) number of pairs for the simulation of currents in nanoscale aggregates. The prediction of electronic couplings can therefore be useful in simulations of the electronic properties of molecular materials. This is a fundamental enabler for higher scale simulations which, enforcing these "individual" results, can offer results on a higher level of aggregation, allowing researchers to infer properties and discover new materials and phenomena otherwise impossible to compute at this level of precision, accuracy and scale. As a matter of fact, at the actual state of the art, is very difficult if not impossible to use actual, computed properties of a lower scale (for example: the molecular or atomistic level) to compute the properties of higher scales (for example, a big bulk of a material or the interaction between two or more different materials).

The prediction task

In ordinary simulations, we need to compute the electronic coupling explicitly between all selected pairs of molecules. However, within this formalism, the electronic coupling depends on the configuration of the dimer only, in terms of atomic positions of the two molecules involved. In other words, the relative position of all atoms in the two molecules is the only information needed to compute the electronic coupling. We may therefore think about

a learning/prediction task, where the relationship between the configuration of a molecular pair and the resulting electronic coupling is first learned, on the basis of computed data, and predictions are made for arbitrary molecular pairs. The full information about the configuration of the molecular pair is, however, quite complex. If we have M atoms in each molecule, the exact representation of a configuration of a molecular pair requires $2 \times (3 \times M)$ reals for the 3D coordinates of the two molecules and $2 \times (M)$ integer values for the atomic numbers. For molecules constituted for example by 100 atoms, we therefore need 600 reals and 200 integers, leading to a quite complex representation. Moreover, the direct use of cartesian coordinates is usually not recommended to represent the relative position of objects in space, as they are not translationally and rotationally invariant[56]. We therefore need to find a more efficient way to represent the configuration of a molecular pair and use a quantitative indicator (feature) to relate the configuration to the coupling.

2.3.3 Descriptors of molecular pairs

Molecular systems can be considered as objects in 3D space. A pair of molecules A and B can therefore be defined in terms of two (generally different) objects in 3D space.

To make things simpler, we first define the standard translation and orientation of a molecular pair by translating the center of mass of molecule A to the origin and aligning the two centers of mass along the x axis. We stress that we are roto-translating the whole pair, and the relative position of all atoms in the two molecules is therefore unchanged. This assumption is valid for any molecular system. In principle, this operation leads to a unique definition of the configuration of a molecular pair, thus removing the issues related to translational/rotational invariance. However, for an exact representation of the configuration of the molecular pair, we would still need $2 \times (4 \times M) - 1$ values if using cartesian coordinates[4].

[4]Here, M is still the number of atoms of the molecule.

We will therefore try to use approximate descriptors of the configuration of molecular pairs, which are able to correlate configuration with resulting electronic coupling in an accurate and efficient way.
Intermolecular coupling depends generally from the intermolecular distance. A good option can therefore be to include, in the set of descriptors, a measure of the intermolecular distance.
In the particular case considered, the intermolecular coupling depends strongly on the mutual orientation of the two molecules involved. Moreover, in amorphous compact aggregates, the structure of molecules deviates quite significantly from the ideal (vacuum phase) structure. Approximate descriptors can therefore include:

- Descriptors of the intermolecular distance: even in a bulk, single molecules are somewhat recognisable. This fact is very important for computing the properties of a bulk of molecules, whose properties are determined by an aggregation of the properties of the individual molecules or from the properties of specific smaller molecular aggregates (like in this case with molecular pairs). In particular, in this experiment we need to be able to find a way to measure a general concept of distance between molecules. This measure correlates quite well with the electronic coupling, as known in literature[100].

- Descriptor of the intermolecular orientation: The mutual orientation between two molecules can first be defined in terms of the orientations of each molecule constituting the pair. If we can associate a vector v (in a given space) to the orientation of a molecule, for a pair of molecules (A, B), the mutual orientation between the two molecules can be defined in terms of the two vectors vA and vB. Associating an orientation to a molecule, however, needs some sort of rules. Moreover, we need a measure of the difference between the two orientations, that is equivalent to finding the distance between the two orientation vectors in the multi-dimensional vector space considered.

- Descriptor of the molecular deformation: molecules have an ideal structure, which is the one that comes from assigning coordinates to each atom by only considering the inner forces of the molecule (i.e. the forces that spawn from the interactions between the atoms themselves). However, in a realistic context, molecules are also influenced by the other molecules that surround them. This extra interactions forces the molecules to change shape, giving birth to what we can define as a deformed version of the same molecule.

2.3.4 Preliminary results

We used two datasets. These datasets are coming from different random sub-sampling of the same DPBIC bulk, so they are comparable since they do not have any physico-chemical difference, but on a statistical standpoint they can show different behavior. In fact, distance mean and distributions are quite different (Fig 2.11), while the rotations are statistically identical[5] (see Figure 2.9).
We tried with a naive approach, using the most basic approach to each of the aforementioned dimensions that we need to measure:

- For the intermolecular distance, the most obvious choice is the distance between the centers of mass of the two molecules in a given pair. In this specific case, since the molecule has a spherical shape and a very heavy atom at its center, we can try to use the coordinates of the two central atoms to compute the distance between the two molecules

- For the mutual orientation, we can try to use the euler angles. Since they are known to have many problems and ill behaviors[56], we can also try to use the more stable quaternion representation or the rotation matrix representation.

- For the deformation, we can use the RMSD. The acronym RMSD stands for Root Mean Square Deviation (also known as Root Mean

[5]We performed a t-test which resulted in a 0.8 p-value.

Square Error), which is a very common way of measuring the difference between N values (i.e. sample or population values). It is usually used to aggregate the magnitude of the errors in predictions for various data points into a single measurement, but in the realm of computational chemistry it is also a very good way of measuring the per-atom distance between two entities (like molecules). In particular, since we are using it on two structurally identical molecules, it can give us the relative amount of distortion between two "real world" molecules or between the ideal version of the molecule and a distorted "real" one.

We then selected a bunch of different models to train with this data, in particular we trained a Kernel Ridge Regressor (KRR)[57, 169] and a Gradient Boosting Regressor (XGB)[55]. These have been chosen both for their ease of use, expressiveness and flexibility.
For all these models, we used the same approach during the training phase:

- We chose the specific features for the training (one for each one of the physical characteristics listed above)
- We chose the tuning strategy and parameters[6]
- We trained each model (with the same tuning strategy) on both dataset separately and then on the combined dataset

However, this first experiment did not turn out to be successful. None of the selected models managed to learn to fit the problem at hand.
Despite the tuning, both **XGB** and **KRR** achieved a **MSE** of around 0.03 (against the normalized Coupling, which has been transformed using the log_{10} and then normalized in the range $[0, 1]$ using the a min-max scaler[7]) when

[6]For the **KRR** we tuned on a grid comprising five kernels (namely the *RBF*, laplacian, polynomial and *linear* kernels) and α ranging from 1.0 and 10.0 with a step of 1.0; for **XGB** we tuned on a grid comprising the number of estimators (testing in particular for 10, 50, 100 and 500), the learning rate (testing for 0.0001, 0.001, 0.01, 0.1 and 1.0), the subsample (0.5, 0.7 and 1.0) and the max depth (testing for 3, 7 and 9)

[7]As implemented in the ***scikit-learn*** library

using both datasets. We also tried to split the datasets into two subsets, one containing only the molecular pair with low distance (under or equal to 13 Angstrom) and one with the long distanced ones (above 13 Angstrom); we tried this approach in order to make the problem a little bit simpler, trying to fit only a local relationship (and so more feasibly fitted as a liner relationship) between the geometrical features and the coupling value. However, even this approach proved to be unsatisfactory.
A visual representation of the results of such fittings can be seen in figures 2.7 and 2.8.

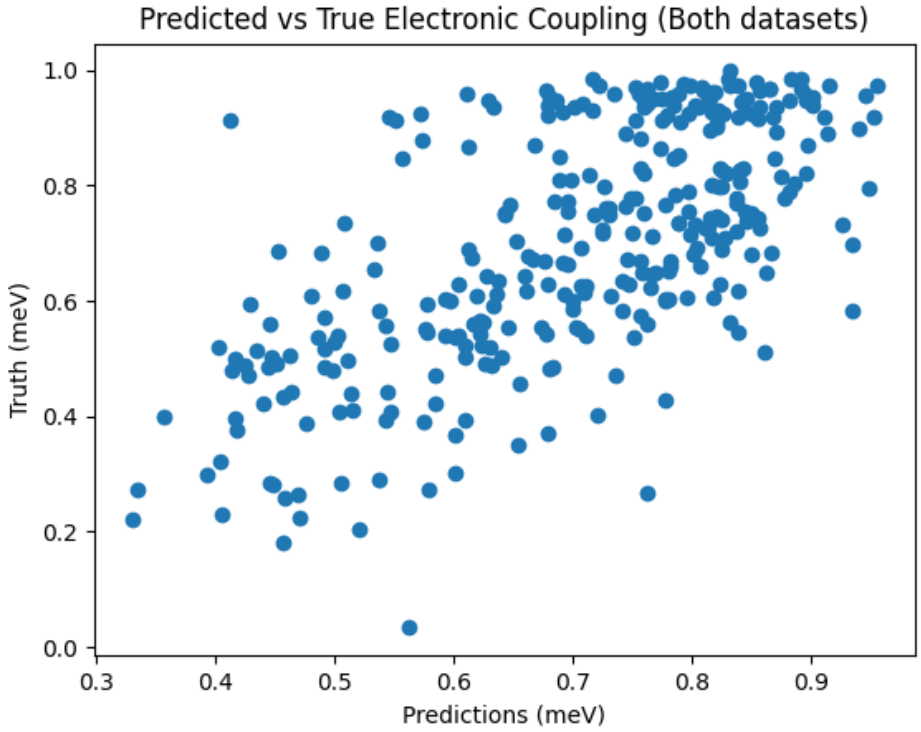


Figure 2.7: Plot depicting the prediction performance of a Kernel Ridge Regressor on the both DPBIC pairs datasets combined.

Our first interpretation was that the dataset was too small to allow us to use it to train a model, but some of the aforementioned previous works

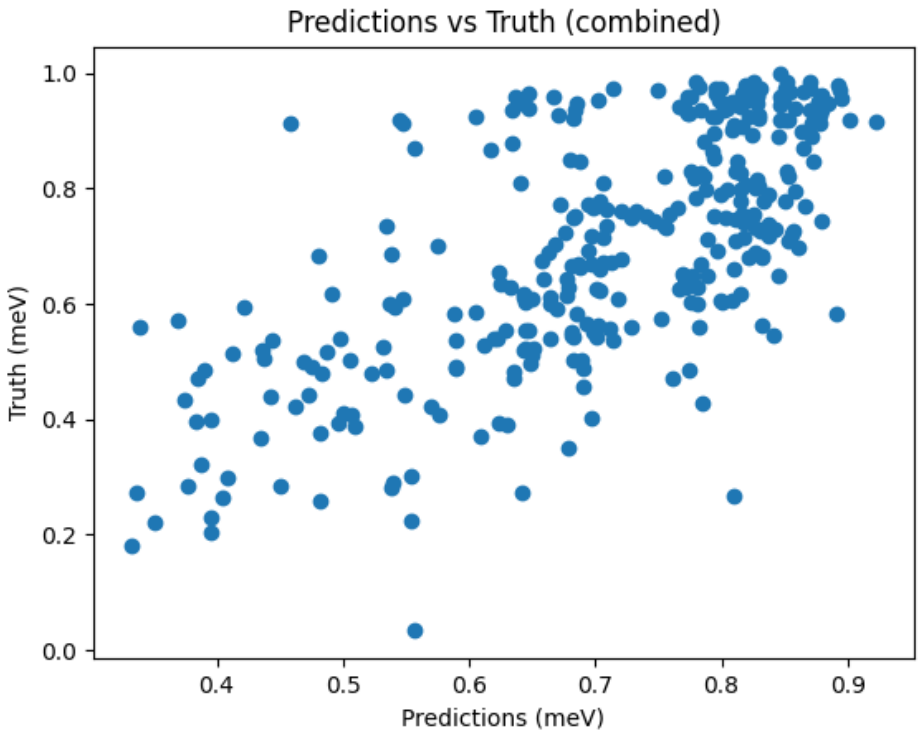


Figure 2.8: Plot depicting the prediction performance of a Gradient Boosting Regressor on the both DPBIC pairs datasets combined.

managed to use datasets of the same size. So, this left us with two other hypotheses:

1. The complexity of our problem (in particular, the peculiarities of the molecule selected) has stricter requirements than those in the highlighted literature

2. Our dataset, other than small, does not sample the latent space in a sufficiently uniform way, meaning that some parts of the possible configurations are not enough represented in the dataset, making it impossible for the model to be able to learn to predict the corresponding

value[8]

To check if the second hypothesis is true, we plot the distribution of different values: the electronic coupling itself, a single number representing the distance between the two molecules and a single number representing the rotational distance between the two molecules. These plots are visible in figures 2.9, 2.10, 2.11 and 2.12.

These figures (in particular figure 2.12) give us good hints of the fact that our models were not able to correctly learn the relationship between our features and the electronic coupling because of the uneven sampling of the latent space in our data. As in many cases with non-fitting models, there is a great probability that the data used are not enough or, more probably, that they do not sample the entire latent space sufficiently or uniformly. This problem is even more common in cases where a part of the latent space is more interesting/important (for example, we might be more interested in having a lot of data for the dimers which exhibit high electronic coupling) but the data have way more information about the least interesting part. This problem is known in literature as ***unbalanced domain***.
However, it is not trivial to figure out how to solve this problem, since it is hard to come up with realistic molecular pairs with the characteristics needed to balance our datasets, and leading a simulation campaign able to overcome this limitation could prove to be very long, expensive and inefficient[9].
Moreover, we are not sure that this is the only reason why our first approach failed. For these reasons, we then developed a simplified system which is able to emulate the behavior of our real-world scenario, but giving us more control on the whole system and giving us the possibility to generate a high volume and high quality dataset with ease. This process is explained in the next section.

[8]Moreover, it could be that these undersampled portions of the latent space are the most interesting ones (i.e. those with the highest or lowest electronic coupling)

[9]Sure enough, due to the intrinsic variability and chaotic nature of molecular materials facbrication (even in a simulated context) we can end up performing a lot simulations and still getting a strongly imbalance dataset.

2.3.5 Building an artificial model system

One of the major limitations we encountered is the fact that many data sources related to chemical entities (aside from those on single molecules) are extremely small or unspecific. For this reason, we decided to design an ideal model for which we can easily generate a great amount of data but maintain affinity with the real case.
Starting from the actual geometry of the real molecule introduced before, we decided that a sphere with a set of marked points on its surface can be used as a model for a molecular system for which intermolecular coupling can be computed. The intermolecular coupling can be considered as the coupling between the two sets of points on the surface of two neighboring spheres, and can be simply computed as an exponential function of the distance between the points.

Data generation procedure and rationale

We want to be able to create a set of N pairs, generated by random orientations of two spheres, for which we calculate the coupling. This will subsequently be used for training a ML model.
The dataset will contain:

- Geometric coordinates for the mutual position/orientation of the two spheres, (quaternion coordinates, rotation matrix, euler angles, RMSD, ...)
- Computed (simulated by a simple point-to-point exponential function) intermolecular coupling.

The steps to generate the dataset are:

- Define the position of a reference set of points (the "coupling" elements) on the sphere surface[10]

[10]This is done randomly, using the Marsaglia algorithm.

- Generate two random quaternions
- Rotate the points of the two spheres
- Translate the two spheres with a distance function (fixed, random within a given range, etc.)
- Compute coupling
- Recompute the features and coordinates[11]
- Generate a dataset list (with the geometric features, quaternions, etc.)
- Convert it to a dataframe and write to file
- Visualize correlations

Model description

A pair of spheres in 3D can have i) different orientations with respect to a reference orientation and ii) different relative positions of the centers of mass (top panel). This is equivalent to fixing one of the spheres in a reference position and with a reference orientation (middle frame) and considering the position and orientation of the second sphere. In this case, the only information needed about the system pertains to the second sphere (since the first sphere is always in the same position).
Alternatively, we can align the two centers of the spheres along a reference axis (bottom panel). In this case, the relative rototranslation information about the two spheres is defined by i) the scalar distance between the two spheres ii) the rotation of the first sphere iii) the rotation of the second sphere. Note that in this latter case the rotation of both spheres is needed. We cannot simply rotate the first sphere to a reference position and rotate

[11]While this step may look redundant and useless, it helps us to make the process as realistic as possible, giving a more sound comparison with the real-case scenario even from a computational load perspective and allowing us a nearly 100% code reusability.

the second sphere accordingly (see Figure 2.13).

Therefore, the most convenient way to represent the relative orientation and position of two spheres in 3D space is translating the whole system (the pair) in order to align the first sphere with a reference position, and considering the translation and rotation of the second sphere as coordinates. We need to perform two tests: one just rototranslating the second sphere randomly, and a second one where both spheres are rototranslated randomly, and the whole system is translated to align the first sphere to a reference position and orientation.
The common setting is the following:

- The reference point is at (0,1,0) on the surface of the spheres.
- The first sphere is fixed at (0,0,0) and with a fixed orientation.
- The center of mass of the second sphere is at a fixed distance with respect to the 1st sphere (2.0) and at a random position.
- The second sphere is randomly rotated with respect to the first sphere.

As for the previous example, we directly use the quantities used to generate the rotation as features.

2.3.6 Testing the Model system

Here we apply the whole ML and data pre-processing pipeline introduced in sections 2.3.3 and 2.3.4, applying that to the model system we developed in section 2.3.5 and see how it performs. Since we have more control on the data generated, we can test if the inability to fit the real system was due to the model and features or if the problem relies in the data available.

Features distribution

The range of the features seems to be a critical point for fitting. The fitting may be affected by the non-uniform distribution of features sampling

a uniform space of rotations.
We first tested the distribution of rotation points (random rotations) obtained from two different methods (Marsaglia method and renormalization, respectively). Results are very similar in the two cases. We can therefore assume that the distribution of rotations is sufficiently random. Within this assumption, ideally we want to use features that are uniformly distributed for a uniform sampling of rotations.
For a distance of 5.0 and a very large number of points (i.e. 100.000), we obtain different distributions for different features. In particular, the translation vector components and the rotation matrix elements exhibit a normal distribution, while Euler angles, axis-angle components and quaternion components are distributed in a less regular fashion. It must be noted that also the artificial coupling is not uniformly distributed, but shows a Gaussian-like distribution.
Figure 2.14 shows some of these distributions for the most meaningful features.

Translation vector components are uniformly distributed, as expected. Therefore, we can conclude that translation vector components can be used safely.
For efficient learning, translation vector components need to be renormalized.
Quaternion components are NOT uniformly distributed for a uniform sampling of SO(3). This is known in literature[178]. The relationship between uniformly distributed numbers and quaternion components is:

$$h = (\sqrt{1-u_1}\sin 2\pi u_2, \sqrt{1-u_1}\cos 2\pi u_2, \sqrt{u_1}\sin 2\pi u_3, \sqrt{u_1}\cos 2\pi u_2)$$

Therefore, quaternion components are distributed as the *sin* and the *cos* functions. This may lead to issues in learning.
Interestingly, the matrix components are uniformly distributed. This can be crucial in learning algorithms.
Regarding the other minor features:

- Position components are more or less random.
- The components of the axis-angle representation are distributed in two different ways: uniform distribution of the axis component and cos-like distribution of the angle. This may lead to inaccuracies, similar to those we get with quaternions.
- One of the components of the Euler angles is also not uniformly distributed (the pitch angle, in this case).

Thanks to this analysis, we can conclude that the use of a normalized translation vector and a rotation matrix can therefore be considered the best representation of relative roto-translation.

Symmetries

When dealing with multi-point scenarios, the asymmetric case performs much better than the symmetric one. At first glance, this would mean that we need to "symmetrize" the coupling in symmetric systems. In other words, the learning could get confused when different possible orientations lead to the same coupling. Let us consider for example the simple 2D case of two circles with coupling points, represented in figure 2.15. One of the circles is fixed, and the other one can rotate around its center.

If we have a single coupling center (black lines in 2.15), the coupling is a function of a "rotation coordinate" q that is, in this case, a measure of the rotation of the circle. The domain of the variable q is, in this case, $[0, 2\pi]$. The coupling has a maximum at 0 and 2π and a minimum at π.
If we have two symmetric coupling centers, however, the coupling has a different periodicity (red curves in 2.15).
However, what we are trying to do is to learn the function $f(q)$ that reproduces the correct behavior $f(q) = c$. If we are able to correctly represent rotations through the coordinate q, we should be able to learn any arbitrary function $f(q)$, irrespective of periodicity etc. However, numerical issues can be the cause of the fact that the specific shape of $f(q)$ can affect the accuracy

of learning.
So, we try to use a description of roto-translation of spheres that uses the "minimal" rotation coordinate. Essentially, we consider the three vectors for each sphere and compute the rotation matrix that represents the minimal rotation that maps one sphere onto the other, computed as a quaternion distance. We also need to consider all contributions to coupling, as we are "symmetrizing" contributions by exchanging the order of points.

Results

At the end of this evaluation, we then evaluated all the approaches systematically. As expected, the rotation matrix proved to be the best representation for the mutual orientation of the two spheres, while the translation vector has been used as the feature containing information about mutual distance. However, even other features (like the quaternion) proved to be useful in fitting this relation. The plot below (Figure 2.16) shows some of the results for different features set[12].

This ideal experiment gave us some insight about the best ways to actually represent our real-world problem and some possible ill properties of our data that can hinder the learning process of our ML algorithms. In the next section, we are going to analyze how we applied this new knowledge to the original problem.

2.4 Back to the real data

Thanks to these promising results, we gained some confidence in the fact that our approach to the problem is solid. That means that the reason why the model cannot fit on the real-world data is not related to our featurization or model selection, but was actually caused by the ill behaviors shown in the

[12]It must be noted that only the best sets have been reported

data. The next step has been the use of specific statistical techniques for data-augmentation that are also able to re-balance the dataset distribution.

2.4.1 Dealing with unbalanced domains

With *unbalanced domain* we refer to a dataset where the distribution of classes is not equal. For example, in a binary classification problem, if one class has significantly more examples than the other class, the dataset is considered unbalanced. This can cause issues during the training and evaluation of models, as the model may be biased towards the majority class and have difficulty accurately classifying the minority class. Unbalanced domains are common in real-world applications, such as fraud detection or medical diagnosis, and require special consideration during the modeling process.
As the above description suggests, this problem has been extensively studied for classification problems, with well known statistical techniques like:

- **SMOTE** (**S**ynthetic **M**inority **O**ver-sampling **TE**chnique)[22]: it works by selecting a random sample from the minority class and computing the k-nearest neighbors for that sample. Synthetic samples are then generated by taking the difference between the selected sample and its k-nearest neighbors, and multiplying that difference by a random number between 0 and 1. The synthetic samples are then added to the original dataset to balance the class distribution. **SMOTE** is a popular technique for handling class imbalance because it can increase the diversity of the minority class while also retaining the original distribution of the minority class.

- **ADASYN** (**ADA**ptive **SYN**thetic sampling approach)[54]: quite similar to **SMOTE**, the main difference between **ADASYN** and **SMOTE** is that **ADASYN** adapts the synthetic sample generation process to the density distribution of the minority class samples. It does this by assigning higher weights to minority class samples that are harder to classify, which are the minority samples that are farther from the deci-

sion boundary. This way, **ADASYN** generates synthetic samples that are more likely to be misclassified, in order to make the classification task harder and more balanced. As a result, **ADASYN** is more effective than SMOTE in handling imbalanced datasets with a large degree of overlap between the minority and majority classes.

When dealing with regression problems, the first challenge is how to adapt this kind of approach to work in contexts where we do not have categorial (i.e. discrete) classes of target but we are instead interested in predicting continuous values. The first consideration to make is that, in continuous problems, "imbalanced data" means that the data points have a skewed distribution across the continuous target variable.
The most relevant solutions are:

- **SMOTER** (**SMOTE** for **R**egression)[142]: as the name suggests, this is an adaptation of the SMOTE algorithm. The main difference is that, instead of sampling classes, **SMOTER** is based on the concept of relevance functions[163]; in a nutshell, those functions are designed to set a sort of threshold to define classes" between the continuous target variable (i.e. values that have a relevance function above threshold are the relevant cases, while those with a relevance function below the threshold are the irrelevant one). Once the dataset is divided, the least represented region(s) are oversampled in the same way as SMOTE does, using a linear interpolation of the real targets to create the new, synthetic one.

- **SMOGN** (**S**ynthetic **M**inority **O**ver-Sampling Technique for Regression with **G**aussian **N**oise)[17]: this technique is a direct improvement of **SMOTER**, which also adds Gaussian noise to the synthetic samples after the interpolation phase. The addition of noise to the synthetic samples helps to increase the diversity of the synthetic samples, which can improve the performance of regression models on imbalanced datasets.

It must be noted that these methods can also be used as data augmentation techniques, helping us to overcome the fact of having a restricted amount of data. We then added **SMOGN** in line with the aforementioned pre-processing pipeline and then re-trained the model.
The results are reported in figure 2.17, figure 2.18 and figure 2.19.

The whole pipeline, including pre-processing, data balancing and augmentation and the actual training phase took just a few minutes on a normal laptop. The final augmented dataset included around 1200 entries, and we used around 200 of these entries for the test set. The time needed for the testing phase is in the order of 10ms, which is a huge improvement compared to the simulation workflow, which requires around 40 minutes on a HPC cluster to compute the electronic coupling of a single pair of the same material.

2.5 Applying the same pipeline to a different molecule

To further evaluate the quality of the whole workflow developed during these activities, we performed another experiment using a very different molecule in order to see if our model would be able to learn its properties without modifications. Moreover, we also asked the model to predict the energy values of the **HOMO** and **LUMO** of the molecule. It must be noted that these are properties of a single molecule, and while related to the electronic coupling they are obviously not exactly overlapping nor one can perfectly deduce one of these using the coupling or viceversa. If we manage to fit this new context with no or few modifications to our approach, it would be a strong indication of the generality of our workflow.
More precisely, we chose the 5,12-Dihydroquinolino[2,3-b]acridine-7,14-dione molecule (also known as quinacridone)[62], which is a molecule used for organic colorants[94] but that is also known for its semiconductor properties[173].
Its structure can be seen in picture 2.20.

Due to the simple shape of this molecule, we managed to get very good predictive performances with a very restricted dataset both for the **HOMO** and the **LUMO** energies. Figures 2.21 and 2.22 shows the relationships between the predicted values and the actual values of these properties.

Another key element of this final experiment is that we wanted to verify how hard it is, for the general researcher, to use our solution. For this reason, we enrolled an external researcher with no previous experience in programming, ML and data science. After being instructed on the basics of these fields, we asked this person to use our software to predict the **HOMO** and **LUMO** energy of pairs of quinacridone molecules. In a few days, this researcher managed to run the whole pipeline, obtaining the aforementioned results. This is another crucial benefit of choosing simpler approaches based on shallow models and encoded physical knowledge instead of complex models based on generics representations: allowing researchers that are new to the field of data science or even computational science to easily enforce the power of machine learning models with little to no training, furtherly increasing the ease of access to these tools and helping to make them more popular and widespread.

2.6 Discussion

In this first experiment, we showed how investing in a strong featurization process, using knowledge emerging from the published literature, alongside with a good pre-processing pipeline allows us to obtain good results in the prediction of fairly complex properties using simple models and a very low computational power. In the DL era, the hustle of working on hand-tailored features may look surprising or useless at first, but digging deeper in the history of the field of materials science and computational simulations it is easy to see how developing the lightest and fastest solutions is a key enabler for empowering a new wave of multiscale materials design with the ability to merge information relative to different scale in real-time.

Moreover, we also showed that well known statistical procedures for data augmentation can be used in the context of natural sciences, generally improving the quality of the learning process and allowing the application of learning techniques even with relatively small datasets.
Another advantage of this process is the fact that linear models are way easier to explain than deep and complex neural networks. This allows researchers to easily inject their formal knowledge of the problems to the algorithm, giving a new proof of previously known results. Moreover, this also gives them the possibility to enforce the opposite process: using the results obtained using ML to gather new knowledge about the system analyzed which can be relatively easily understood and put to test. This can be done in two different ways:

1. If a set of features is completely unknown: implement a very powerful DL model using unspecific features, in order to find answers to the problem. Then, trying to identify the most relevant characteristics (for example, studying which cases are better or worse understood by the powerful model), turn them into actually usable features and fit a simpler model. If this process is successful, researchers can then be able to find a formal and provable scientific explanation for a phenomenon.

2. If a macro-set of features is known, but it is not easy to understand which are actually relevant: using the whole set to fit a simple model and evaluate the solution found. If the solution is good enough, researchers can then try to progressively remove less promising features and see if the new trained model is as performant as before. Iteratively doing this process (which can be at least partially automated), researchers can find the minimal set of features that are needed to actually solve the problem, and then proceed to use this new knowledge to find a formal scientific explanation to the problem.

Another key advantage of the second approach is that it makes way easier for researchers coming from different backgrounds to understand and use the

resulting ML models, due to the simplicity of the resulting software and to the adherence of this software to their pre-existing knowledge. We showed this fact in our last experiment, easily leading a researcher with no programming and computational science experience to run our software in order to fit a machine learning model able to predict slightly different properties than the one we used and for a completely different molecule.

At the actual state-of-the-art, this whole process is hindered by a main factor, namely the lack of bigger databases, able to collect results obtained from different teams, using different softwares (and, consequently, encoded using different formats). This leaves the ML practitioner with a choice: using small datasets, relying on data augmentation procedures (which, however, can prove to be insufficient for very small datasets used for more difficult problems) or trying to merge different datasets, investing a very long time trying to understand the structure of all the different formats, with the possibility of finding out (potentially after a long period of time) that not all file formats contains the same information, and then being forced to use other statistical procedures to infer the missing features or not using the incomplete features at all.

This problem needs to be addressed, leading to the creation of common data platforms. However, as easily understandable from what stated above, this is not and easy task; researchers need to understand the information that are explicitly reported together with those that are intrinsically contained in each data format currently used, developing a unique way to encode all of them and in the best way possible, both in a human-understandable and efficiently machine-processable and storable way. In a plethora of different fields, these problems have been historically faced through semantic technologies and, in particular, through the development of ontologies. Indeed, ontologies are the base ingredients for the creation of knowledge graphs, which can then be used to create formal specifications for databases using the intrinsic relationship between entities used in that specific field of knowledge.

These technologies and their application to our research activities are going

to be the main theme of the next chapter.

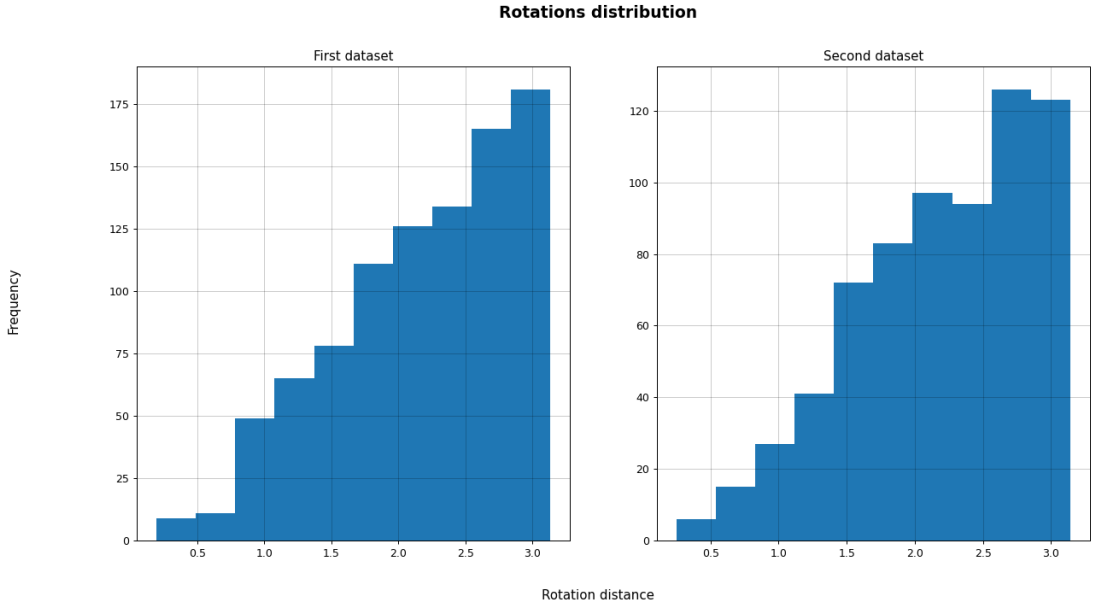


Figure 2.9: Two plots - one for each dataset - depicting the distribution of the rotational difference between the molecules in the pairs. To plot this in a single figure, we transformed the corresponding rotation matrix for each pair into a numerical value which gives a measure of the magnitude of the rotation needed to align the two molecules. In particular, this transformation is the following: $\theta = arccos((trace(M) - 1)/2)$. As we hypothesized, the vast majority of the pairs falls in a specific part of the latent space, which is the one with a high difference in orientation. As known, these are the pairs with the lowest coupling, hence the least interesting ones. The t-test resulted in a p-value of 0.815.

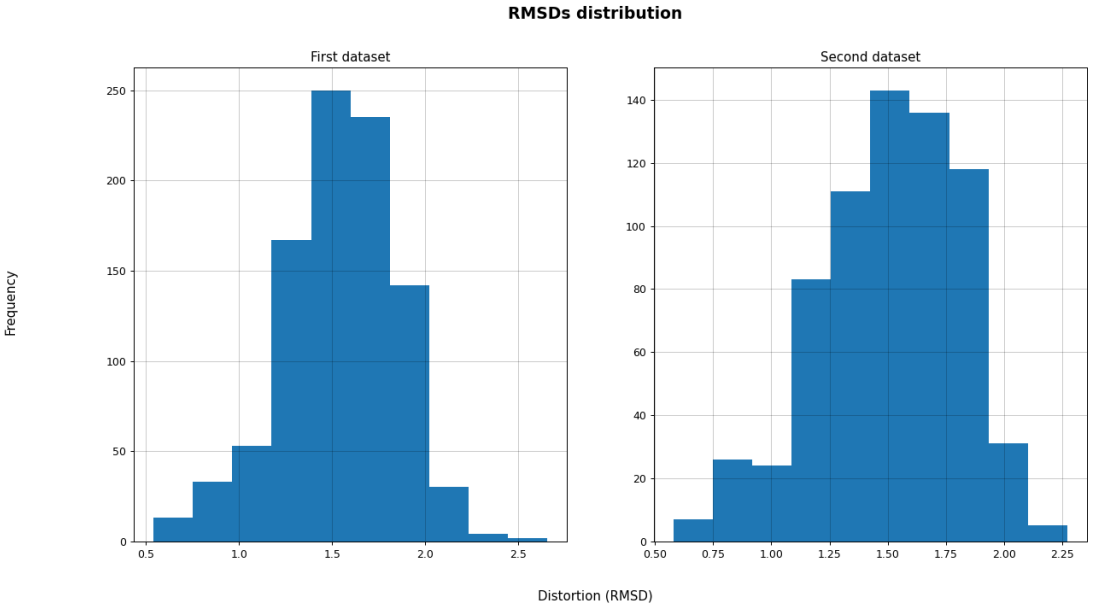


Figure 2.10: Two plots - one for each dataset - showing the distribution of the relative distortion between molecules in a pair. Differently from the rotation matrix histogram, it looks like the RMSDs are roughly normally distributed, which should not be a pathological situation. However, even though this is still an open question, it is very likely that RMSD is a less useful feature than the rotation matrix, so it probably cannot compensate for the ill behavior shown in figure 2.9. The t-test for this two distributions resulted in a p-value of 0.049.

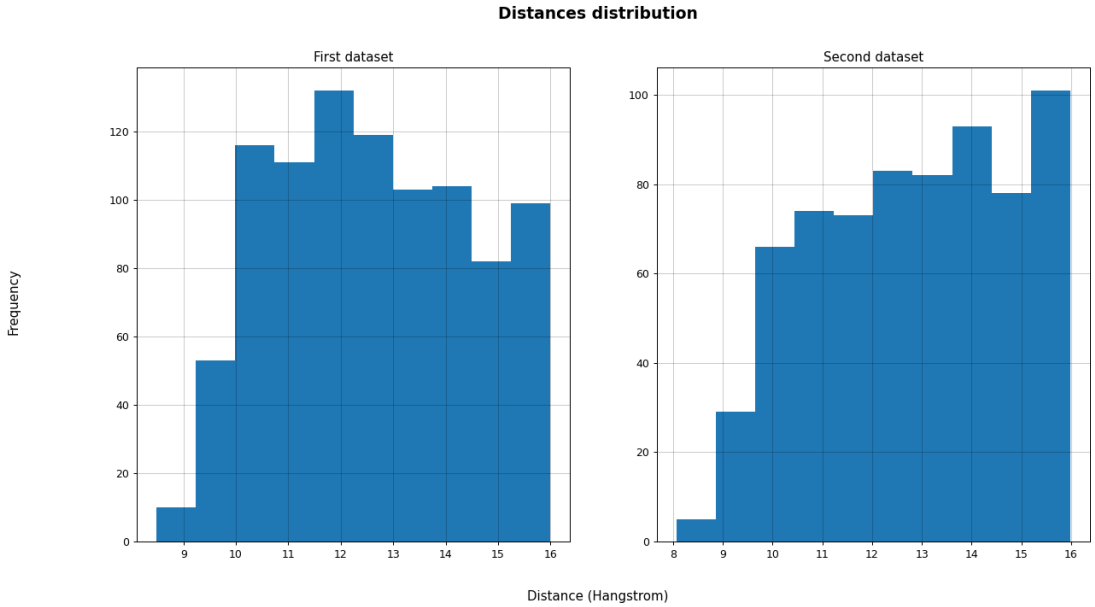


Figure 2.11: Two plots - one for each dataset - showing the distribution of the distances of the two molecules in the pairs. These look nearly uniformly distributed, which is definitely a good scenario, in all the bins other than the two smaller ones, i.e. 9 and 10. However, these are the most important ones, since it is known that closest molecules are the ones exhibiting higher couplings. The t-test resulted in a p-value of 0.017.

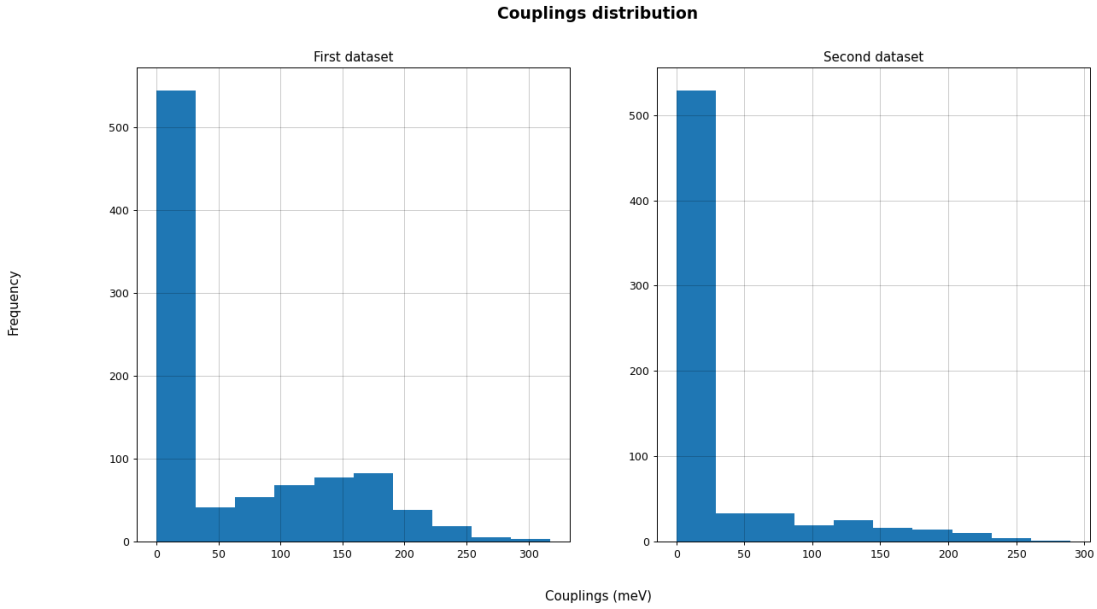


Figure 2.12: Two plots - one for each dataset - showing the distribution of the electronic coupling of each molecular pair. Both are very skewed towards the left, probably due to the combined effect of the ill behaviors shown in figure 2.6 and figure 2.8. This is probably the greatest cause of the poor fit of our ML models, which are probably learning to assign a close-to-0 electronic coupling to nearly all the possible cases. The t-test resulted in a p-value of 3.83^{-19}

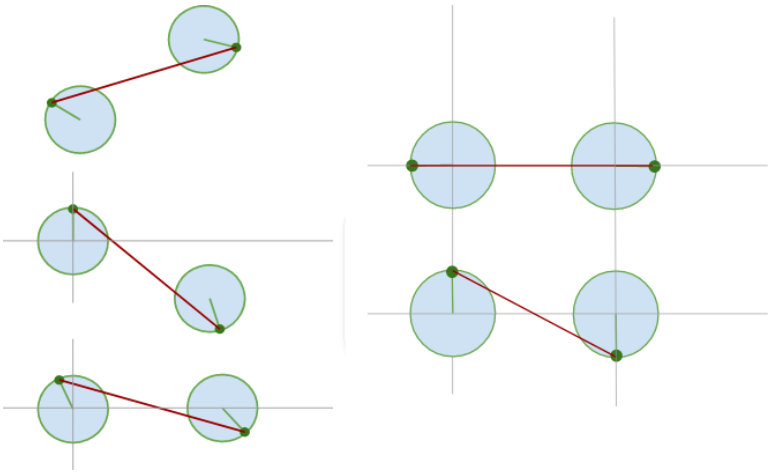


Figure 2.13: Some examples of mutual orientation of two spherical objects with aligned centers. In a) we can see positions that would be representable using only the rotation of the second sphere; in b) we can see a mutual orientation in 3D (upper image) that when transposed on aligned spheres requires both rotations to be represented.

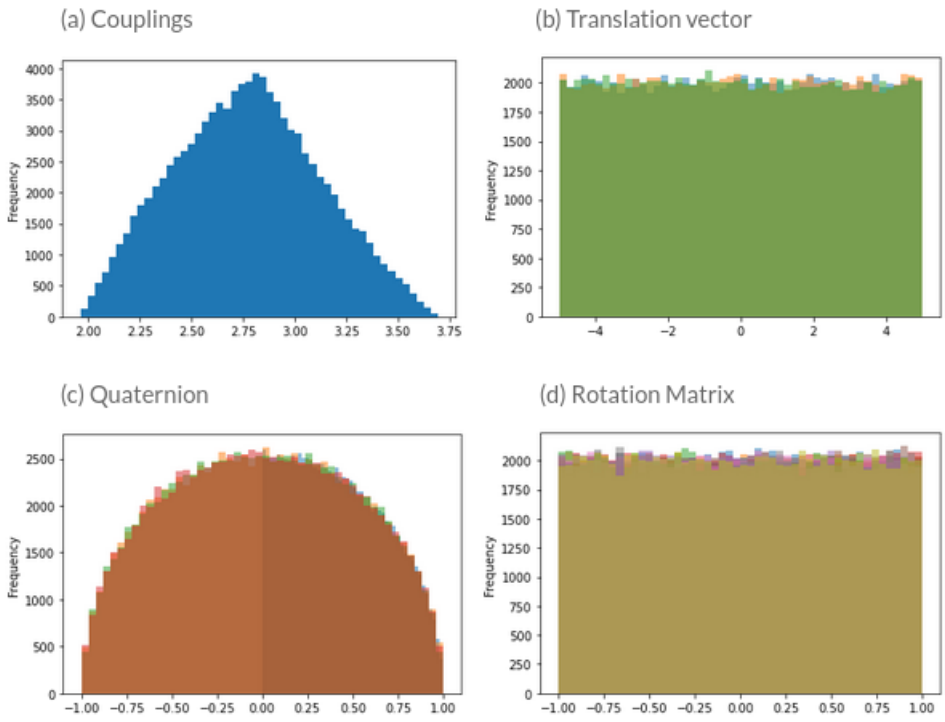


Figure 2.14: Plots showing the distributions of the main features. We can see how the components of the translation vector (b) and of the rotation matrix (d) are uniformly distributed, while those of the quaternion (c) are not. Coupling (a) is somewhat normally distributed.

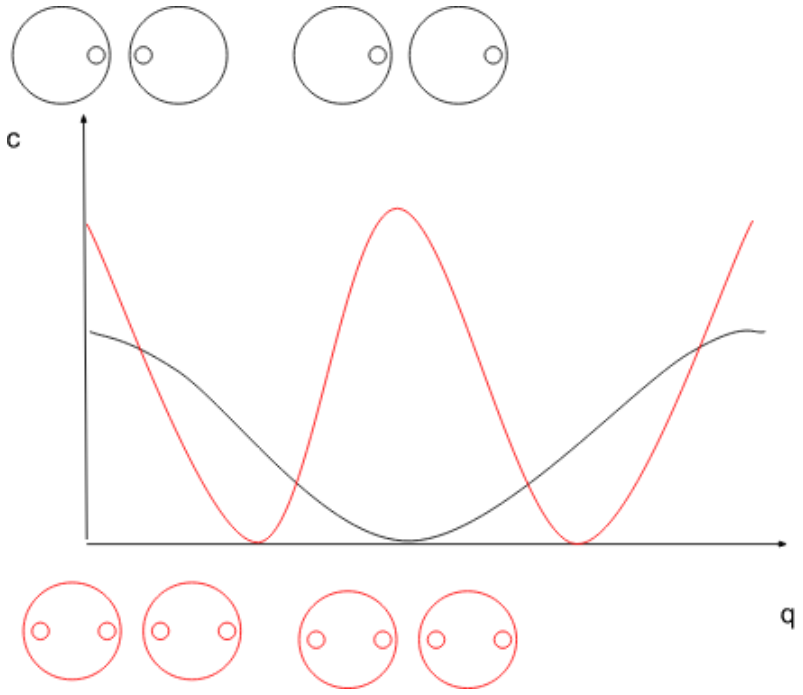


Figure 2.15: A plot showing the trend in the value of the artificial electronic coupling depending on the position of the two spheres. The black line shows the trend when the spheres have only one marked point, related to the position of the two black spheres drawn over the plot; the red line shows the same trend when the spheres have two marked points, and the corresponding spheres position is drawn under the plot. It is easy to see that the two-point scenario deeply suffers from the symmetry problem.

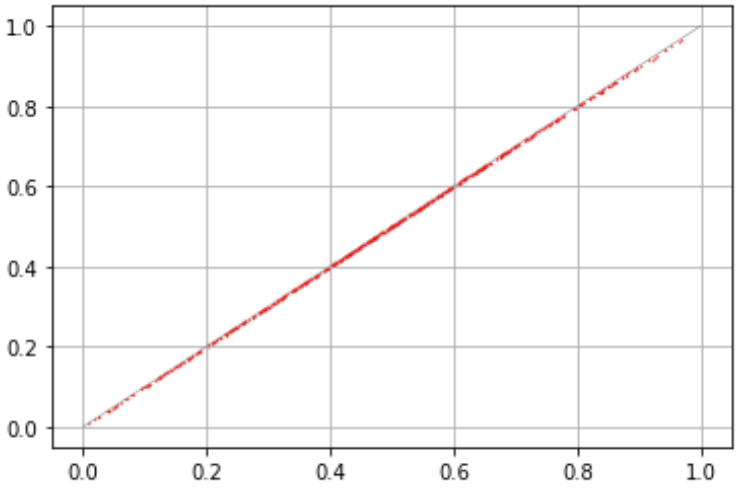


Figure 2.16: A plot showing the performance of the trained model in predicting the artificial coupling using the distance vector and the rotation matrix as features. On the x-axis there are the predicted values, on the y-axis the actual values of the coupling.

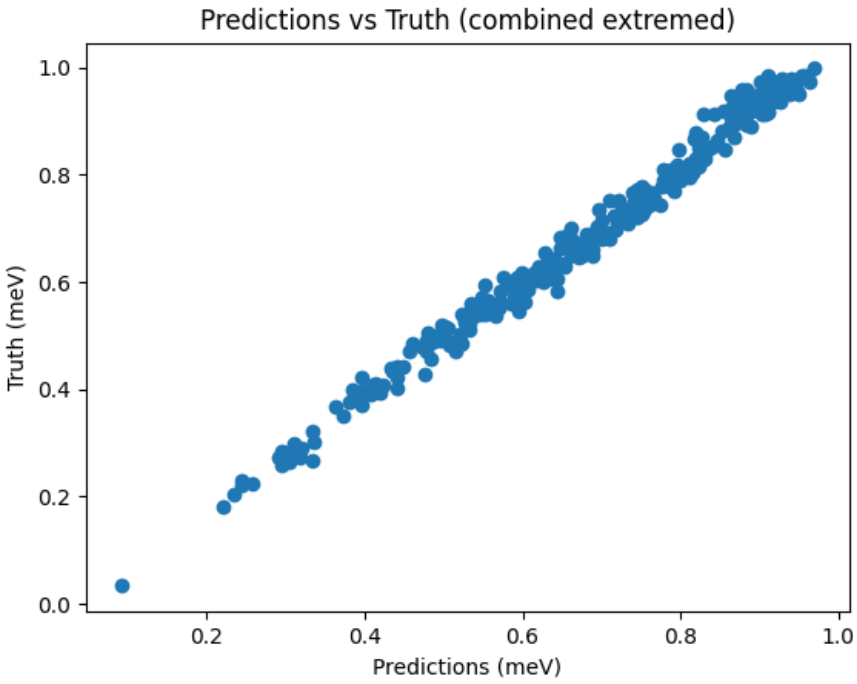


Figure 2.17: A plot showing the relation between the predictions made by the **KRR** model (on the x axis) and the values obtained from the simulations (on the y axis) using the distance vector and the rotation matrix as features. Other than appreciating the vast improvement against figure 2.7, we can already see how the resampling algorithm allowed us to have a dataset which samples the latent space more uniformly and, even more importantly, that contains more values on the important side of the spectrum (high coupling values) instead of low values (near zero values, nearly absent in the plot).

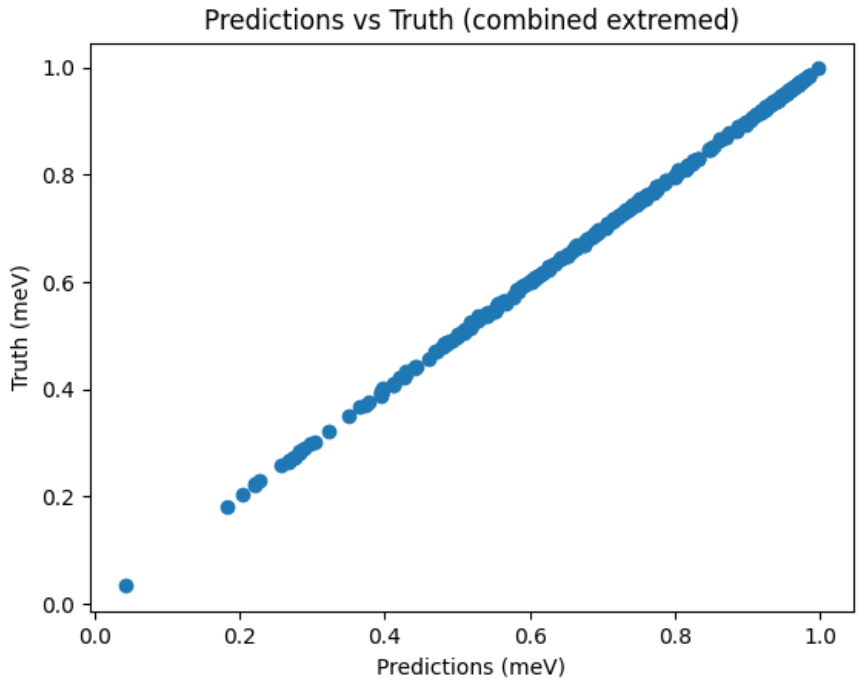


Figure 2.18: A plot showing the relation between the predictions made by the **XGB** model (on the x axis) and the values obtained from the simulations (on the y axis) using the distance vector and the rotation matrix as features. Other than the nearly perfect fit, we can already see how the resampling algorithm allowed us to have a dataset which samples the latent space more uniformly and, even more importantly, that contains more values on the important side of the spectrum (high coupling values) instead of low values (near zero values, nearly absent in the plot).

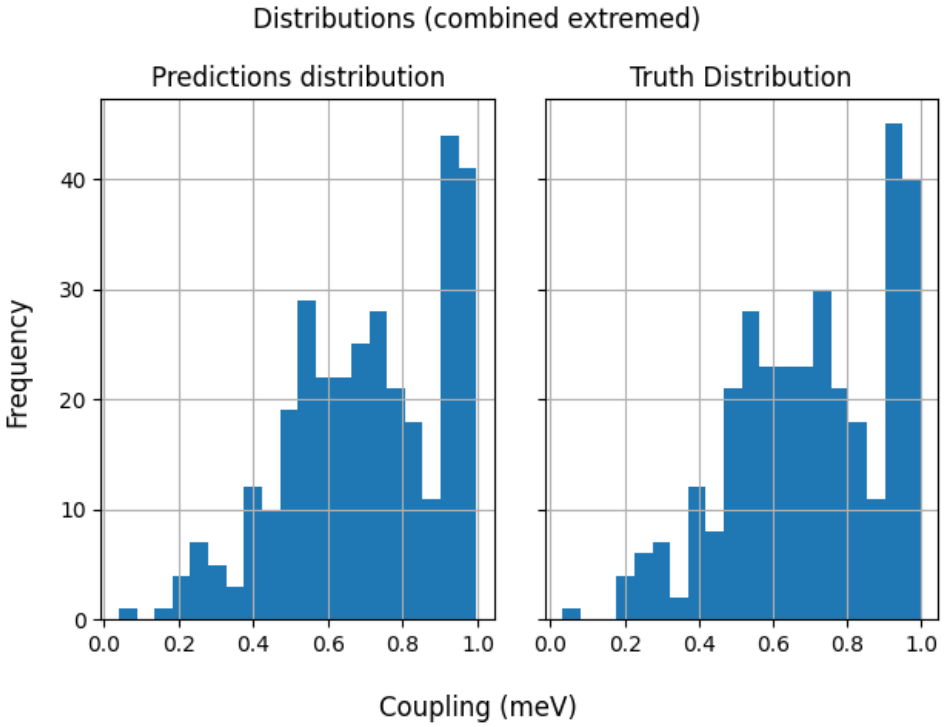


Figure 2.19: Two histograms showing the distribution of the values of the electronic coupling predicted by our **XGB** model (on the left) and the values given by the simulations (on the right). Again, these are results obtained using the distance vector and the rotation matrix as features. We can see how the distributions are actually quite similar (the t-test resulted in a p-value of 0.96, confirming that the null hypothesis cannot be rejected). Moreover, like in figure 2.18, we can see how our dataset now contains more entries with high values than entries with lower values. The two distributions are statistically equivalent, as proved by the t-test which results in a p-value of 0.998.

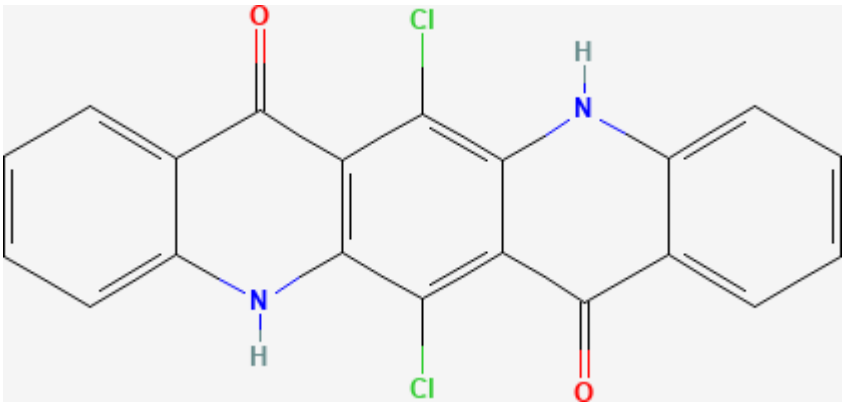


Figure 2.20: The 2D chemical structure of the quinacridone molecule.

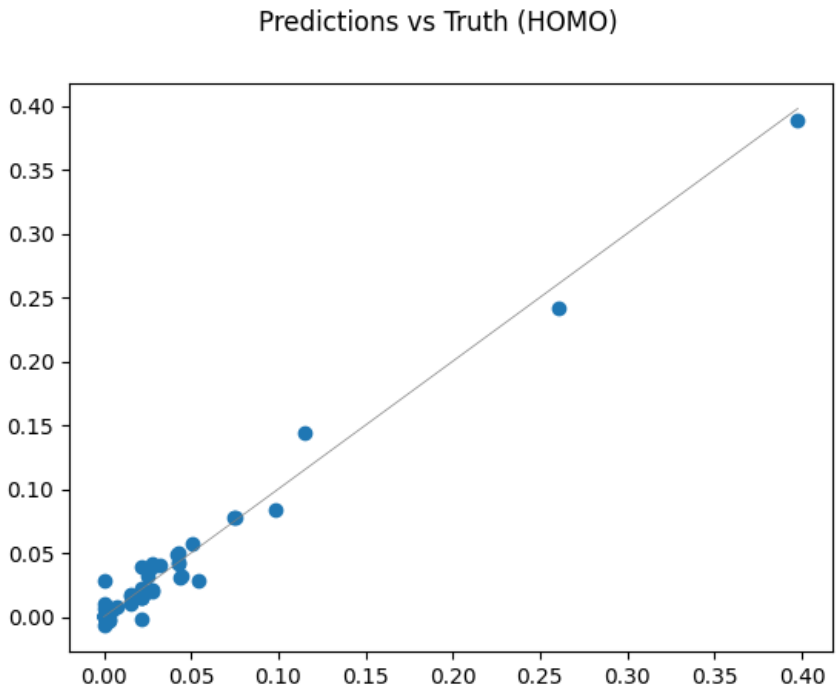


Figure 2.21: Predictions against truth for the HOMO energy of quinacridone pairs.

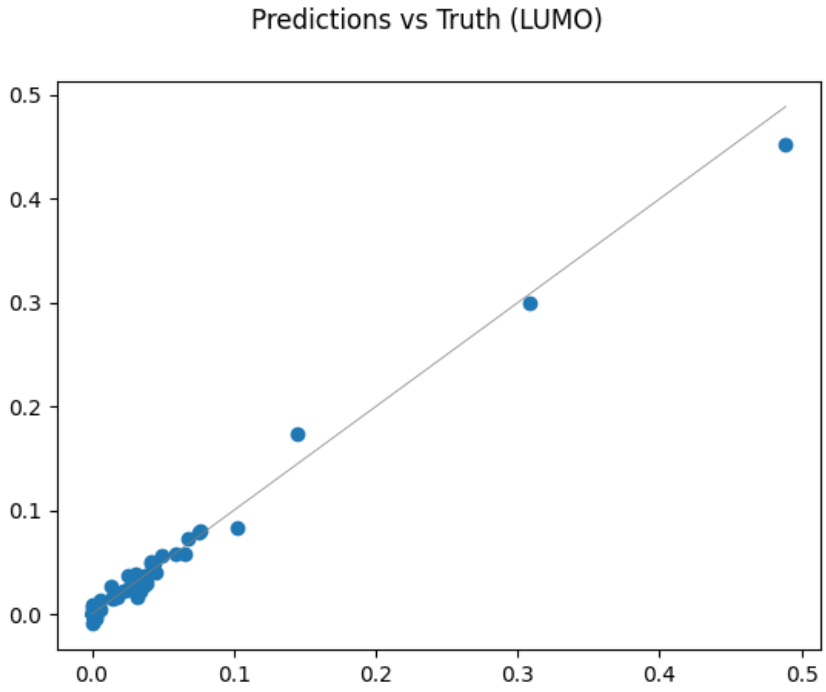


Figure 2.22: Predictions against truth for the LUMO energy of quinacridone pairs.

Chapter 3

Semantic Data Management

In the previous chapter, we introduced two major limitations of the current state-of-the-art in the field of materials science R&D activities: the lack of systematic approaches to deal with features and descriptors, and the huge effort needed for data generation. But we had to face many other challenges during our research activities: a huge plethora of different software and tools to perform simulations and measurements (and the consequent proliferation of data formats), the lack of platform well suited to share data related to chemical entities, the lack of data formats able to offer the right amount of information for each different application and sub-field and the inevitable necessity of representing the same entity in many different ways depending on the specific activity performed. These peculiarities are part of the intrinsic complexity of the field, which requires to be treated carefully and in a systematic fashion in order to be embraced and, progressively, solved.
To overcome these limitations, we need a system that is able to represent the entities we manage in our day to day work into a full-fledged database. This would strongly improve many aspects of computational and experimental research activities dealing with materials and chemical entities in general: easy data access and retrieval, common formats for easier sharing and less dispersion, etc.
However, standard database tools are unable to effectively solve these prob-

lems, since the objects and procedures used in materials science are not easily represented using classic database tables. Moreover,the standard material scientist is not an expert programmer or database user, and relying on queries written in standard programming languages, with the classic programming structures is probably not a winning solution, together with the fact that possible very useful queries are not easily expressed in standard programming languages even by experienced programmers.

Many of these problems share a lot with the discipline of semantic web, which strives to create a formal representation of all the world wide web, make it completely machine readable and, consequently, more easily parsable, giving the possibility to offer users way more powerful tools while enforcing natural language[128].

One of the main tools used for the semantic web (and semantic technologies in general) are ontologies. Ontologies can be seen as a set of concepts that define a realm of knowledge, paired with the relations between them, which togethers allows researchers to create a formal model of their research domain. Using this model, we can create a knowledge graph, which is the object that links the actual information in our hands using the concepts and relationships we defined in the ontology. Enforcing specifically tailored technologies like **SPARQL**[136], we can then query our data using the concepts and the relationships we defined.

An additional issues is modelling workflow in a systematic way. Ontologies can also be exploited for modelling workflows providing the terminology used in defining tasks, problem-solving methods (PSMs), and domain definitions. The accomplishment of a task can be realized by applying problem solving methods. Here, we use PSMs for the structuration of knowledge and evaluate terms and concepts in our ontology[46, 118].

We wanted to focus on a real use-case. The realization of the task is then defined in terms of methods, which are also related to pre- and post-conditions. The workflow can be defined as a pre-composition of methods or in an incremental way (on-the-fly).

We need therefore to define tasks (goals and methods) and sub-tasks, as well as methods and pre-conditions. The instances of pre-conditions define terms that must be associated with the ontology terms. Further, relationships between concepts/instances can also be formalized, for this specific case.
In this chapter, we are going to analyze all the activities related to ontology development and semantic technologies that we brought on during the last three years, starting from our custom ontology for the materials science domain.

3.1 General context

The general domain of this ontology is that of materials science, and in particular advanced (nanostructured) materials. With that expression, we mean materials specifically designed to have specific properties, to accomplish specific goals etcetera. Moreover, the domain extends to the end devices built with such materials, and the goals of such devices could be the actual reason for the development of the materials itself. By "nanostructured" we mean the materials exhibit peculiar structural and morphology features at the nanoscale, and the "shape" of the materials can be studied in terms of morphology and interactions of their constituting units, for example clusters, molecules or atoms. Speaking of end devices, another problem arises which is that of interfaces, which (roughly speaking) is the zone where the bulks of two different materials come into contact.
Materials science is composed of many different realms and the overall workflow is complex and composite. It ranges from measurements, the actual fabrication of the devices, many different processes for many different materials, all kinds of experimental procedures and, of course, theoretical analysis. Obviously, there is another kind of research that plays an important role in materials science, namely computational modelling. The general workflow can and usually include many stages, mixing activities stemming from both approaches. For example, an experiment can lead to mixed results, and to

make further investigation easier researchers can decide to organize a simulation campaign in order to avoid to loose time due to environmental effects and the other unpredictable elements introduced in chapter 1. After the simulation campaign, researchers can extrapolate new elements and then go back to a fabrication and measurement phase, which can be the last part of the whole workflow or can lead to new phases, both experimental and computational. This continuous shift, the exploitation of numerous techniques, tools, softwares, machines and the respective conventions of both the experimental and computational fields, leads to a plethora of different languages, data formats and naming rules. To this general framework, we hope to add the exploitation of data-driven methods (in particular machine learning) in order to propel research in this sector and to accelerate the discovery of new materials and devices. We believe that data science and machine learning could be able to not only accelerate the entire process, but also to give researchers the ability to create something that could not be discovered without the implementation of such techniques.

It is easy to understand how the complex workflow highlighted above, and the combination of the numerous elements that are necessary to build both experimental and computational activities introduces a lot of complexity, and being able to retrace the entire research process that led to a specific result becomes rapidly impossible without a common way to describe the entire workflow in all its steps and together with the results of each of these steps. However, such a standard is still missing and the actual state of the art often consists in handcrafted scripts, input files and codifications/representations which then have to be described with ad-hoc articles, repositories and so on. That means that nearly anytime a researcher tries to replicate or even understand and enforce the results of another one, she has to deep dive into the logic and the ad-hoc choices made to carry on the specific study she is interested in. This is not only a waste of time, but also a major impediment to open science and to the ability to quickly gain new insight and knowledge from previous results. Moreover, the probability of making mistakes is

strongly incremented.

The goal of this project is the realization of a unique entity able to give researchers a common way to represent, record, lead and share not only the final results of an experiment, but also the entire process, tools and scientific reasons that led to such results.

To serve as an example, we can think of a workflow similar to that introduced in chapter 2.3.3 for OLEDs. There, we enforced multiscale modelling and machine learning, but we could have also performed experimental measurement after our simulation and ML campaign. In particular, we can imagine our team of researchers identifying a promising molecule, then simulating the charge transmission of the corresponding molecular material. Such a property (charge transfer) does not depend only on the molecule but also on the process we use to actually "create" the final material. In order to identify the topology of the best possible material, we then need to calculate a different property at a different scale, namely the electronic coupling between two molecules. Such pairs must be extrapolated from the starting bulk that we already simulated, then we have to calculate the target property for many (potentially all) the possible pairs, then we can start to reason about which could be the best topology on a statistical standpoint. Then, another problem: obtaining the electronic coupling for a single pair may need 40 minutes on an optimized computing cluster, and a bulk is made of thousands of pairs. Here, machine learning can play an important role, because a trained agent, once trained, can give us the same property in milliseconds. Once obtained the electronic coupling of a sufficient number of couples, we can then use a statistical method to see if our suggested topology is actually better than the one we started at the beginning. At this point, we may go back to the fabrication laboratory, fabricate the resulting promising material and test its actual performances in the real world, dealing with all the chaotic elements typical of the fabrication process (1.3).

In order to efficiently go through such a workflow, we need all the features and commodities we introduced before, starting from standardization of data

structures, the ability to easily retrieve valuable molecules for a given target (in this example, charge transfer) and, if someone already led an analogous research, we should be able to easily access to the corresponding results, in order to start from where other researchers already arrived. Moreover, defining standard nomenclatures and languages, jointly with common data structures and formats, can be a fundamental key in providing more interoperability between experimental and computational results, further empowering the joint exploitation of both realms.

3.2 Ontology Development

The **MAMBO** approach arises from a plethora of practical needs that exist in the domains related to the development and application of advanced materials, particularly molecular materials. One of the most pressing issues in this field is the need for a structured and standardized method of designing and executing R&D activities, as well as representing materials data, information, and workflows. With a focus on providing practical solutions for common research-oriented activities, the **MAMBO** development approach is centered on the definition of a specific set of relations, concepts, and tools that enable researchers to communicate with each other and standardize their activities, enabling more interoperability between different methods and research teams. Additionally, this approach should also furnish researchers and experts with tools for optimizing and automating settings and procedures.
To meet these goals, the **MAMBO** development approach incorporates concepts from the Problem-Solving Method (PSM) approach. PSMs are useful tools for implementing task-based frameworks in Knowledge Engineering (KE) and can link the representation of knowledge and information to operational tasks[162]. This makes PSMs an ideal tool to use in the **MAMBO** development process. Accordingly, the **MAMBO** approach is guided by the creation of a framework where PSM tools specific to the domain are supported by a domain ontology, providing terms and concepts for specifying

tasks, data, and workflows[46, 118]. This allows for a more comprehensive and accurate representation of materials data, information, and workflows, which is essential for effective decision making, uptake, and utilization of materials modeling by a wide range of manufacturing end-users.
Definition coming from previous work[43] describe PSM as built by three basic components:

- Competence: the description of the input and output behavior related to a given task, together with a description of what the PSM can achieve.
- Operational specification: the description of the reasoning process which links the required knowledge to the specified competence previously defined.
- Requirements/assumptions: the description of the domain knowledge needed by the PSM to achieve the competence. Simply put, requirements and assumptions describe the pre-requisites needed by the inference steps described by the operational specification for the application of the PSM to achieve a target.

It comes that PSMs are a tool that can be used to formally define tasks and their solution (and perform the same solution) enforcing the application of domain knowledge. It implies that a PSM requires two ingredients: the task to complete and the actual knowledge of the domain. Then, the task execution can be described as a sum of methods, which are connected one to another by pre- and post-conditions. PSMs allow researchers to define complex workflows and operations using pre-constituted recipes or with an incremental (on-the-fly) composition of simpler methods[43]. In this work, we applied these general principles behind PSM approaches to the development of **MAMBO**. **MAMBO** is intended to enable researchers to perform tasks and relative subtasks, organizing the knowledge required to formalize them and that is gathered after their execution. This brings us to a mutual relationship between the ontology, **MAMBO**, that should be able to provide all

the concepts and relations needed to formalize the knowledge required to apply PSMs, and the PSMs themself, which requirements are used as the basic ingredients used to give birth to the ontology. For these reasons, **MAMBO** aims at being a "lightweight ontology"[43], whose target is to help solving practical use-cases, organizing the required knowledge, rather than being a basic and all-encompassing ontology for materials science and molecular materials. By utilizing this approach, we have the ability to concentrate on the practical, real-world scenarios that arise from research within the specific domain. By using the categories and relationships established in **MAMBO**, we can form methods that pertain to individual tasks, breaking down complex workflows. Additionally, by continually incorporating new concepts and relationships discovered through the examination of various applications and issues, we can continuously refine the ontology to better address and solve those real-life tasks.
To develop **MAMBO**, we have to clearly define the typical tasks and subtasks, as well as the methods and pre/post-conditions associated with use-case scenarios within the chosen domain. The pre-conditions will be linked to terms within the ontology. Additionally, connections between concepts will be established to address specific situations. A thorough examination of the application scenarios and use-cases is essential for the successful development of **MAMBO**.

3.2.1 Related works and ontologies

First of all, we led an analysis of the current literature about ontology development related to materials science domain. This makes us come to contact with pre-existing ontologies that influenced the development of **MAMBO** (and other less related to our goals and work), while also giving us the possibility to notice the lack of specific work on molecular nanostructured materials. After this observation, we started our investigation about common ontology development methodologies. The ontologies that most influenced our work are introduced below.

EMMO The **E**lementary **M**ultiperspective **M**aterial **O**ntology)[42]. Its development is a multidisciplinary endeavor that aims to create a standard framework for representational ontology. Rather than beginning with broad, abstract concepts like other ontologies, **EMMO** is built from the ground up using the actual physical world as understood by physics and materials science. While highly useful as a foundation, **EMMO** is designed to be general and non-specific, and can be used at both the top and middle levels of ontology. Its main purpose is to focus and organize the knowledge of fundamental physics, chemistry, and philosophy behind applied sciences, specifically materials science. **EMMO** defines all the basic building blocks necessary for organizing knowledge in these fields, leaving it to subsequent work to establish functional and enforceable categories and classes for practical applications. From this foundation, various sub-ontologies have emerged, including **MAMBO** which borrows different elements from the general design criteria used during the development of **EMMO**, specifically in the definition of relationships between classes, with the aim of potentially using **MAMBO** as a domain ontology of **EMMO**.

MDO The **M**aterials **D**esign **O**ntology)[88]. It is developed in order to offer a framework that defines concepts and relationships to encompass knowledge in the field of materials design. It aims to create a knowledge representation that can bring together experimental and computational results in a common format and structure. Additionally, **MDO** is specifically designed to enhance information retrieval from databases that store information in the domain of materials science. While **MDO** has the capability to represent knowledge in various sub-fields of materials design, particularly computational tasks, it primarily focuses on crystalline/periodic systems and single molecules. Therefore, it is less suitable for representing and managing information about non-crystalline compounds and materials. **MDO** is constructed upon Competency Questions, as answered by materials science experts, and Use Cases, which establish the purpose and foundation upon

which the ontology is designed. **MDO** is structured as a modular ontology, centered around the core concepts of Structure, Provenance, and Property. The term "Structure" refers to the basic chemical makeup of a specific material, while the term "Property" describes the quantitative and qualitative chemico-physical characteristics of a material. In the development of **MAMBO**, we reused these general concepts defined in **MDO**, and adapted them to the specific context of the target domain. Additionally, we linked specific classes and attributes already defined in **MDO** to **MAMBO**.

DEB The **D**evice, **E**xperimental scaffolds and **B**iomaterials Ontology)[49]. It is an open resource for organizing information about biomaterials, their design, production, and biological testing. It is created using text analysis to identify ontology terms from literature on biomaterials and is systematically curated to represent the domain lexicon. **DEB** can be used for searching terms, making annotations for machine learning applications, standardizing meta-data indexing, and other cross-disciplinary data exploitation. As an ontology for Biological materials, Devices and Experimental scaffolds, **DEB** is closely linked to **MAMBO** in terms of foundations and approach. Additionally, the complexity and heterogeneous nature of the literature and operational approaches in **DEB** are similar to those addressed by **MAMBO**. One of the unique features of **DEB** is the semi-automatic selection of field names to be inserted in the ontology, in contrast to common approaches that rely on domain experts. Furthermore, the **DEB** terms set is dynamic as new elements are added when the need for new terms arises from new research. Ultimately, **DEB** is explicitly oriented towards integrating machine learning techniques with classical computational approaches. Specifically, terms included in **DEB** have been selected using a bag-of-words approach, and the completeness of the selected terms set has been validated in two different phases: first, a curated selection of articles have been mined, to verify if all articles were easily findable with the terms present in the ontology. Then, 70 domain experts have been asked to do the same with their articles and

themes of interest, proposing new missing terms to be added to the ontology. The approach used in the development of **DEB** therefore points towards self-updating ontologies, which can automatically be upgraded with newer terms and vocabularies leveraging some automation and data-mining techniques.

ChEBI The **Ch**emical **E**ntities of **B**iological **I**nterest)[30]. It is an ontology and database for molecular entities that focuses on small chemical compounds. The term "molecular entity" is defined by **ChEBI** as "any constitutionally or isotopically distinct atom, molecule, ion, ion pair, radical, radical ion, complex, conformer, etc., identifiable as a separately distinguishable entity." The molecular entities covered by **ChEBI** include both natural and synthetic compounds that have potential biological activity. The concepts defined in **ChEBI** for the characterization of the structural features of molecular systems are useful tools for organizing knowledge in domains where individual molecules and their components are involved. The **MAMBO** ontology builds upon the **ChEBI** approach to defining hierarchies for molecular systems and subunits by developing concepts related to individual molecules.

CHMO The **CH**emical **M**ethods **O**ntology)[13]. This one is an ontology that details the techniques and procedures utilized in chemical experiments, including characterization methods like mass spectrometry and electron microscopy, as well as methods for processing, isolating, and synthesizing materials such as ionization, chromatography, electrophoresis, epitaxy, and continuous vapor deposition. It also includes information on the instruments used during these experiments, such as mass spectrometers and chromatography columns. The purpose of **CHMO** is to supplement the **O**ntology for **B**iomedical **I**nvestigations (**OBI**)[11] and leverages the extensive maturity of **CHMO**, a related ontology for experimental procedures. To strengthen the structure of **MAMBO**, certain sections of **CHMO** were utilized and incorporated into the ontology, specifically classes and relationships related to the development of experimental materials.

MSEO The **M**aterials **S**cience and **E**ngineering **O**ntology)[51]: it is an ontology which uses the Common Core Ontologies stack[146] and aims to give materials scientists the ability to represent their experiments and data in a semantic manner. **MSEO** intends to support researchers with data management tools that are both human and machine-readable, and easily integrated into other scientific domains.

Moreover, **MAMBO** has been influenced by the following projects.

VIMMP The **VI**rtual **M**aterials **M**arketplace **P**roject[60]. This project aims to be a user-friendly platform that offers easy access to all the tangible and intangible elements necessary for efficient decision making, adoption, and effective use of materials modeling by a wide range of manufacturing end-users. This includes information, knowledge, services, and tools. By providing these resources, **VIMMP** aims to speed up the development and market deployment of new materials. The project has been found to be an effective resource for reusing concepts, structures, and relationships.

OPTIMADE The **O**pen Da**T**abases **I**ntegration for **MA**terials **DE**sign)[5, 4] is a consortium that aims to make materials databases interoperable by developing tools such as a specification for a common REST API. One such database that **OPTIMADE** works on is **NOMAD** (**NO**vel **MA**terials **D**iscovery)[60, 33], which creates, collects, stores, and cleanses computational materials science data computed by the leading materials-science codes. **OPTIMADE** also develops tools for mining this data to find structure, correlations, and new information that would not be discovered through smaller data sets.

3.2.2 Preliminary steps

To adhere to common patterns present in the aforementioned ontologies, we identified a working group of about 10 domain experts, with competences

on computational and experimental aspects of materials research and development, supported by knowledge engineers, and we asked the experts to describe typical operations, processes, objectives and goals related to their research projects, daily routine and tools used, gathering a set of relevant use cases and objectives, which can be summarized as follows:

- Representing knowledge on integrated modelling and characterization workflows for advancedmaterials, processes and related technologies.
- Providing a standard representation of materials modelling workflows.
- Providing semantic interoperability across modelling and characterization tools (for example, modelling software tools, characterization workflows, etc.).
- Providing a basis for the development of tools with search/query capabilities in the field of materials modelling and characterization.
- Connecting with existing knowledge in the materials science domain.

This updated, more detailed list guided all the following development activities. Together with these use cases, the analysis team also gathered a first list of relevant words, which has been iteratively updated during the whole process.

3.2.3 Specification of tasks

Starting from the general use cases, the analysis team identified an initial set of more specific tasks that can be supported by **MAMBO**:

- File the results of a materials modelling activity into a database
- Perform a database query on the details of computational methods used (for example density functional theory, pure ab-initio methods, molecular dynamics, coarse-grained methods, finite elements) to generate materials modelling data.

- Perform a database query from large computed datasets to retrieve specific target structures and properties of materials within a given range.
- Retrieve structured information on materials (including data generation and provenance) to be used in predictive ML/DL models, testing accuracy and effectiveness

Although these tasks are already part of the research activities of several teams within the materials science domain, they often suffer from the lack of standardization, optimization and automation. Conversely, materials science workflows often follow a very tailored and customized set of routines and methodologies. One of the objectives of **MAMBO** is therefore supporting interoperability in the implementation of workflows within the specific domain considered.

3.2.4 Competency questions

Following well established and common ontology development schemes[123], the first draft of **MAMBO** structures was designed following the results of competency questions (CQs)[123]. These are questions given to experts of the field analyzed, and these are used to gather the basic knowledge required, assisting ontology engineers in defining the domain considered[123]. Essentially, the ontology should contain all the relevant concepts needed to answer the CQs considered. The analysis team identified an initial set of typical questions for which the information and organization in **MAMBO** should provide answers, including the following:

- Is the material considered crystalline (lattice-structured or periodically organized)?
- What are the chemical formulae of the molecules involved in the molecular material considered?
- Is the material made of a single molecular component or is it a blend?

- Is the material homogeneous, at the molecular scale?
- What is the composition of materials with computed properties (e.g. computed density) falling within a given range?
- What are the material properties and their values that are produced by a given materials calculation?
- What are the molecular structures of the interface between two given materials for which a computed property (e.g. computed density) falls within a given range?
- For a calculated material property, is the calculation based on ab-initio methods, data driven techniques or both?
- For a calculated material property, what is the actual computational method used in calculations?
- For a calculated material property, which software produced the calculation results?
- What is the value for a specific parameter (e.g., cutoff energy) of the method used for the calculation?
- What are the input and output structures of a materials calculation?
- Who are the authors of the calculation for a computed property?
- Who are the authors of the measurement for a measured property?

3.2.5 A first outline of MAMBO

We chose not to follow a top-down or bottom-up approach, but rather to mix them in order to accurately capture the varied nature of concepts involved. We began by creating a preliminary set of qualitative connections among the first identified terms, then further defined the ontology classes and their relationships in greater depth. Our approach was primarily modular,

starting with core concepts and relationships, and expanding on them to more specifically organize knowledge within the domain.

Core structure

The core of the ontology is composed of entities that are closely tied to the tasks and scenarios mentioned earlier. The central idea in **MAMBO** is the concept of Material, which is connected to Structure and Property, and further linked to Experiment and Simulation. This organization reflects the key terms that frequently appear in the specific domain targeted by **MAMBO**. Each of these concepts is closely linked to at least one task related to the domain we focused on during development. The implementation of **MAMBO** will heavily rely on the concepts and connections defined in this stage, as we will demonstrate later.
More detailed definitions of this basic concepts, as of in a submitted paper on **MAMBO**[132], are:

- **Material**: the most general concept related to a material, which defines a portion of matter with some specific attributes (kind, quality, etc.)[42]. Despite very generic in principle, the range of the specific materials covered by **MAMBO** is narrowed by the set of attributes considered. The **Material** concept is strongly related to the following two concepts: **Structure** and **Property**

- **Structure**: term representing all the concepts and relations connected to the structural characteristics of a **Material**. Sub-hierarchies of the **Structure** concept will specify knowledge related to typical material components within the targeted domain (atoms, molecules, etc.).

- **Property**: concept representing all the chemical and physical properties of a material (mechanical, electro-magnetic, etc.). The **Property** hierarchy is conceptually straightforward and new useful categories representing properties emerging from new and/or neighboring domains can be easily added to the ontology.

- **Simulation**: concept representing all the information related to a computational workflow that led to a specific result, from the scientific motivations down to the actual software and parameters. The **Simulation** internal hierarchy is, together with **Experiment**, the most complex of the entire ontology, as it should be able to represent a broad range of scenarios related to different simulation tools and protocols.

- **Experiment**: analogously to **Simulation**, this term represents all the processes and procedures needed to perform an empirical experiment. Together with **Simulation**, the **Experiment** concept represents the knowledge required to define the procedures providing the value of a **Property**, the characteristics of a **Structure** and/or their mutual link. Consequently, **Experiment** and **Simulation** can be viewed as different categories of methods providing different representations of given physical phenomena.

A crucial aspect of these concepts is their high degree of interconnection, which is a consequence of their strong mutual relationships. These relationships emerge from the typical design and investigation activities related to materials science. A sketch of these concepts and their relation is depicted in Figure 3.1.

Questionnaires

Once we reached this point, we organized a new set of interviews with domain experts. In order to make the process quicker and less controlled by the development team, we decided to distribute an anonymous questionnaire. The questionnaire has been built in order to offer experts the level of detail they prefer, leaving the duty to re-map the emergent concepts and abstractions on the existing **MAMBO** core to the development team. At the end of the process, we found that the vast majority of the terms emerged during this process can be easily mapped onto the already existing concepts, and those that cannot are concepts related to devices and akin topics, which

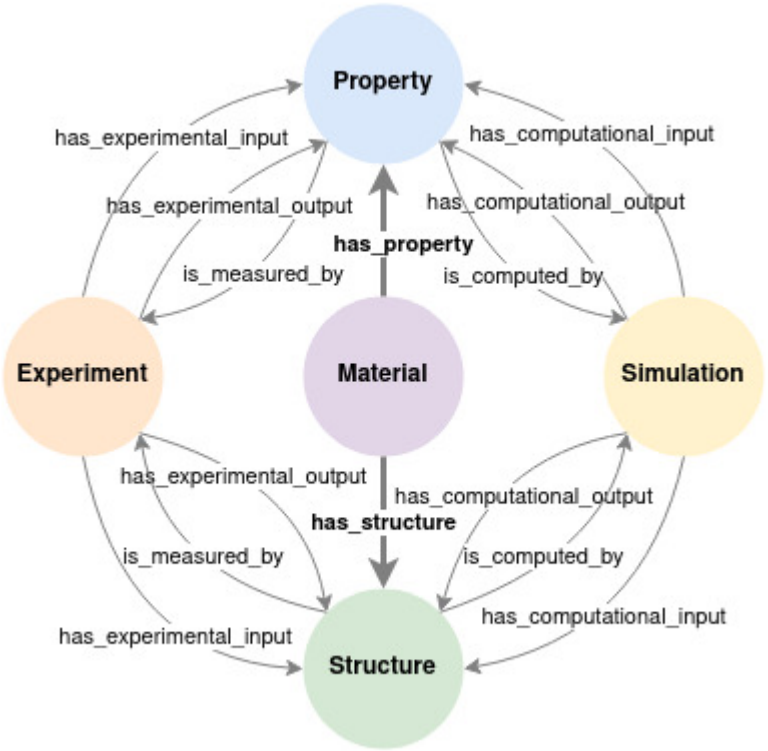


Figure 3.1: **MAMBO** main core classes and relationships[132]: the ontology revolves around the concepts of **Material**, **Simulation** and **Experiment**. An object (**Material**) is represented by its structural features (**Structure**) and properties (**Property**), while computational (**Simulation**) and experimental (**Experiment**) workflows are connected through a common interface to **Property** and to **Structure**

are out of our original scope. The questions are the following[1]:

- Briefly describe your fields of study and interest
- Which is the fundamental component of your work? (Molecule, material)
- How do you usually structure your research process?
- Which of the aforementioned steps is usually the most difficult and time consuming?
- Which role does data have in your research activities?
- How big is the volume of data you usually have to manage? (Some megabytes, some gigabytes, many gigabytes, terabytes, other)
- Is data automatically collected? (Yes, no, partially)
- If partially, explain which steps of the process of data acquisition are automated and which are manual
- Which is the rate of acquisition of data? How often are they collected? How long is this process?
- Redact a list of terms which are needed to speak about your research activities
- If a search engine for materials science existed, which kind of queries would you like to be able to do?
- Fully explain a particularly relevant experiment, starting from the design phase and finishing with the data collection phase

This process lets us focus on many different activities which are being carried out in our institute, and the fact that the structure we depicted so far holds is a good hint of the fact that we captured the underlying structure of the molecular materials realm.

[1]Closed questions have the possible answers reported inside parentheses

3.2.6 Implementation

The whole implementation of **MAMBO** and the concepts described above have been implemented in **OWL**[6] using the **Protégé** editor[121]; the latter is a mature, feature-full and cross-platform software for ontology development. The latest implementation of **MAMBO** is accessible on GitHub[2].

Initially, we formally implemented the concepts and relationships that build the core of the ontology (see Figure 3.1) and then we gradually built more complex hierarchies gradually, progressively covering emerging entities and case studies. The formal definition of the core classes and relationships closely follows those of the concepts introduced in section 3.2.5. Those formal definitions are[132]:

- `Material`: is the main class of **MAMBO** and is related to the `Property` and `Structure` classes via the `has_property` and `has_structure` relationships respectively. An instance of this class represents the "abstract" concept of a given material, and will be related to potentially many instances of `Structure` and `Property`, which in turn can result from different instances of the `Simulation` or `Experiment` classes.

- `Structure`: an instance of `Structure` represents the specific structural characteristics of a specific occurrence of a `Material`, being it a real world physical material or the object of a computational workflow. The `Structure` class is linked to `Simulation` class via the `is_computed_by` relation and to the `Experiment` class via the `is_measured_by` relation.

- `Property`: this class defines the general properties of materials, either computed, measured experimentally or both. Same as for `Structure`, an instance of the `Property` class represents the specific property related to a specific case. Moreover, the `Property` class is linked to `Simulation` and `Experiment` via the relationships `is_computed_by` and `is_measured_by`, respectively, thus in analogy with the `Structure` class.

[2]https://github.com/daimoners/MAMBO

- `Experiment`: this class is linked to `Structure` and `Property` through two corresponding relationships, namely `has_experimental_input` and `has_experimental_output`, respectively. These relationships describe the `Structure` or a `Property` connected to a specific experiment in terms of input and/or the ability to produce a result.

- `Simulation`: this class essentially mirrors the `Experiment` class on the side of computational workflows. The `Simulation` class is linked to `Structure` and `Property` via `has_input` and `has_output` relations.

Deeper hierarchies

These core concepts have been used as the basis for a further structuration of deeper hierarchies for more specific concepts and entities.
In particular, `Structure`, `Simulation` and `Experiment` need a deeper look in order to understand how they work. `Structure` has some attributes like `spacegroup`, `lattice` and `composition`, but the relation hierarchy is strongly based on the the intrinsic characteristic of materials and chemical entities, which in general are composed by many different entities at different scales, ranging from elementary particles and atoms up to molecular aggregates and even other materials. All these entities have been conceptualized on the same level due to the operative nature of **MAMBO**, and are related to `Structure` via the `has_structural_entity` relation (namely: `Atom`, `Particle`, `StructuralUnit`, `MolecularSystem` and `MolecularAggregate`). While the sub-classes inherit all the parameters of `Structure`, the independent classes share some common fields, while storing some specific parameters like `formula`, `atomic_number`, `composition`, `symbol`. Moreover, we introduced classes that will help us in linking **MAMBO** to previous ontologies for materials science; for example, `Crystal` can help us to link **MAMBO** with **MDO**, which is very similar in its philosophy and in many design choices, but focuses solely on materials with a crystalline structure, while **MAMBO** focuses on molecular materials (which usually lack a crystalline structure). However, the ability to enforce both computational and experimental data is

a common trait, and having some modules that directly link to **MDO** will also help us in our future work to more strongly relate to it and to organize **MAMBO** in a way that strongly resemble **MDO**, empowering a more diffuse re-use of terms, relations and patterns. Moreover, we also inherited from **MDO** the concept and class of `Coordinates`, used to describe the spatial placement of a Structure. However, we decided to go deeper using more classes to represent different kinds of coordinates. At the time being, we have a `CartesianCoordinates` class and a `SphericalCoordinates` class.
A depiction of the aforementioned relations is given in figure 3.2.

`Simulation` and `Experiment` hierarchies share many common traits, and this is both a design choice and an intrinsic characteristic of them that we figured out during our preliminary work, during literature review (both in materials science as a whole and in ontology development for materials science) and during our interviews with domain experts. In fact, while methodologies, timing and skills are quite often very different, experimental and computational workflows share many structures and patterns. They both need a unique identifier to be retrievable (namely, `ID`), a text-based attribute for reporting specific information and notes about simulations and experiments (the `log` attribute) and they both have a class of specific methods (`ComputationalMethod` and `ExperimentalMethod`).
Even though we tried to highlight and embrace the similarities between the computational and the experimental workflow as much as possible (following our goal of making experimental and computational data as interoperable as possible) at this point the two hierarchies split and the used approaches start to diverge. `ComputationalMethods` is related to two classes called `Algorithm` and `InteractionPotential`; the first one, `Algorithm`, is used to describe the various different algorithms that can be used to perform the simulations, while the second is used to describe the model chosen for the interaction between the different particles considered. Both Algorithm and `InteractionPotential` both have subclasses; `Algorithm` subclasses are used to represent specific algorithms and optimizers used to perform the simula-

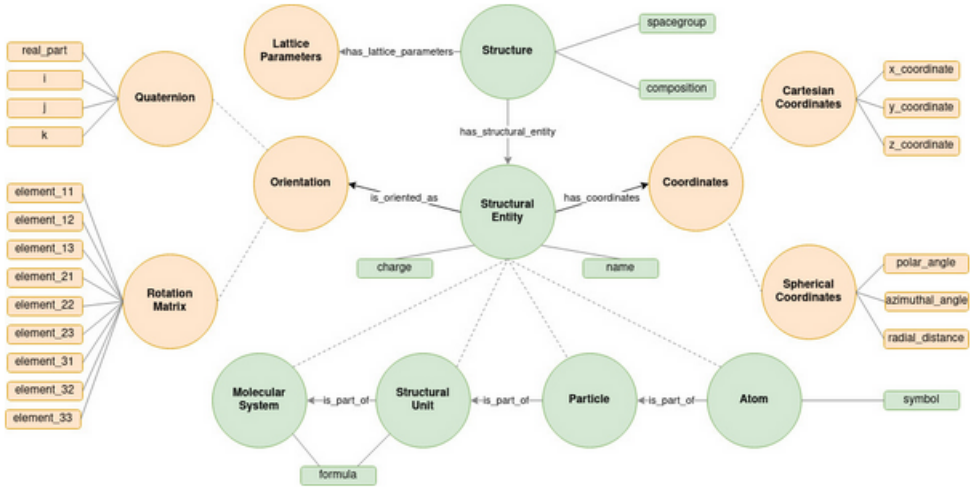


Figure 3.2: Scheme of the `Structure` class[132]. The main concepts and relationships used in the `Structure` class emerge from the analysis of actual workflows in typical problem solving tasks involving molecular materials. Terms and relationships are connected to both computational and experimental techniques and methods. Circles represent classes; squares represent some examples of attributes for the respective classes; dashed lines represent subclass relationships. For the sake of clarity, we omitted `Structure` direct subclasses, for example `MolecularAggregate` and `Crystal`. Green circles represent the classes directly related to structural entities, i.e. those directly related to `Structure` itself and `StructuralEntity`; orange circles represent the auxiliary classes which encode specific characteristics or spatial features, like all kinds of `Coordinates` types or `Orientation` subclasses, information related to the lattice (`LatticeParameters`) and so on.

tion, while `InteractionPotential` subclasses are used to represent different types of potentials that can be used to perform the simulation.
While a similar approach can also be used for `ExperimentalMethod`, we chose not to do that first hand but to exploit the work already present in literature and link that to **MAMBO**. In particular, we decided to import parts of the **CHMO** ontology[13], an ontology specialized in knowledge representation for the experimental scaffold. It already offers a wide range of methods used in the development of materials, and we imported all the hierarchies that contained those methods, linking them to the `ExperimentalMethod` class. In particular, we imported the classes called `continuant`, `continuant fiat boundary` and `process`, together with their respective hierarchies.
Sketches of `Simulation` and `Experiment` hierarchies are visible in Figure 3.3 and 3.4 respectively.

3.3 Using MAMBO in research activities

As already stated, we developed **MAMBO** in order to be used in practical activities related to data management and curation in the molecular materials field. These kinds of activities are already being discussed in literature, and some guidelines and best practices are emerging[48]. The applicability of **MAMBO** in the organization of knowledge in the target domain was assessed by analyzing simple typical workflows related to R&D for materials and in particular molecular materials.
In the following, we focus on simulation workflows for investigations on molecular materials. The analysis of simulation workflows, in particular, allows us to define technical requirements and tune accordingly the expressiveness of **MAMBO** in addressing the specific knowledge involved in the description of materials at different scales (from particles to aggregates). Following the PSM approach, a general workflow connecting initial information and conditions (pre-requisites) and final output (post-requisites) is decomposed into tasks and sub-tasks. The definition of tasks and sub-tasks and the domain

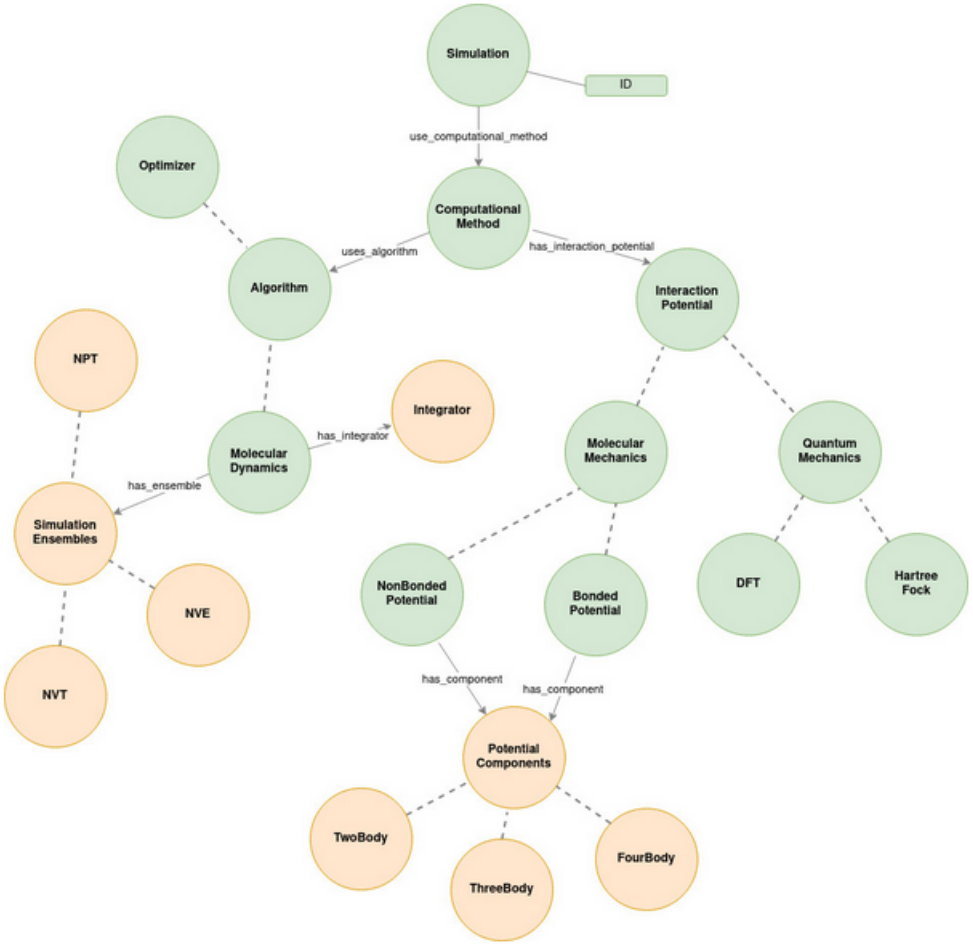


Figure 3.3: Scheme of part of the Simulation class[132]. The **ComputationalMethod** class gathers the different computational methods and their parameters and is related to the Algorithm class (the information about the specific algorithm used for the simulation) and to the **InteractionPotential** class (the information about the inter-atomic interaction potential used in simulations). These classes and their sub-classes are in green. Supplementing classes that collect specific information of algorithms or interaction potentials (e.g., **SimulationEnsemble** and its subclasses) are in orange.

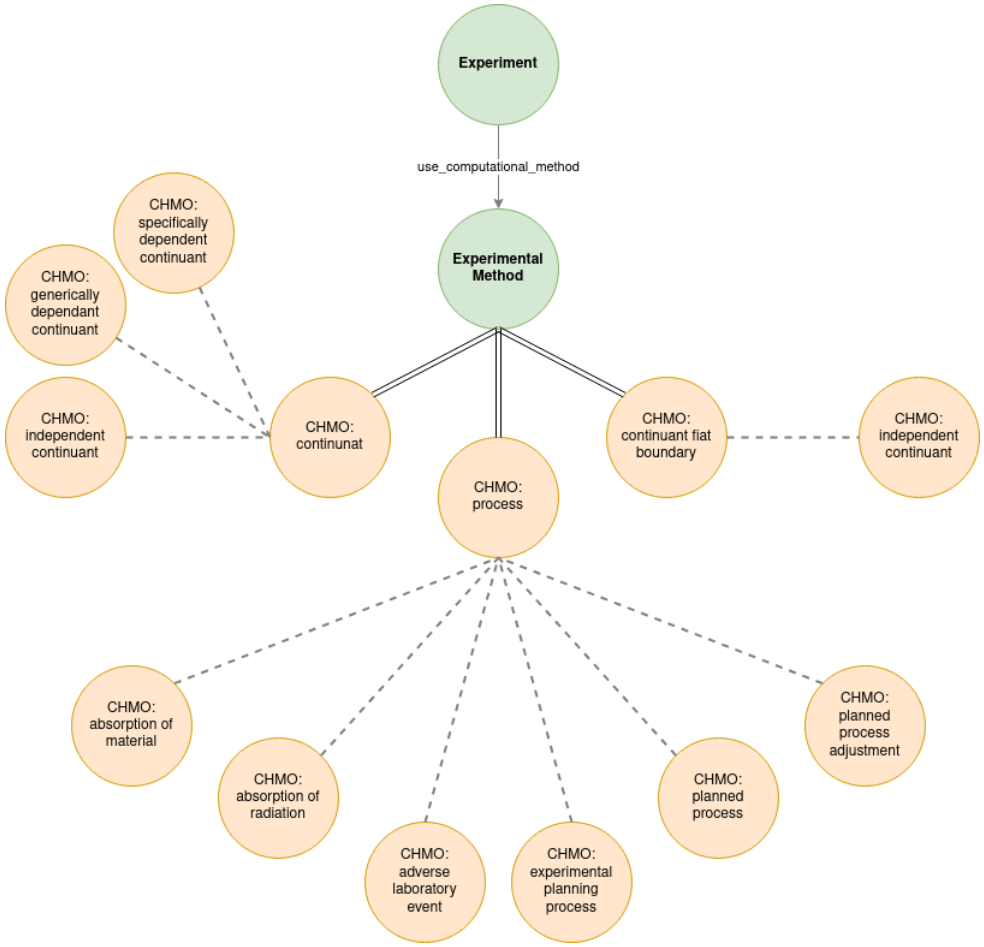


Figure 3.4: Scheme of the `Experiment` class with its subclasses[132]; the green circles represent classes that are original to **MAMBO**, while orange circles represent the classes imported from **CHMO** (this is also stated inside the circle). The double line represents the import relation.

knowledge is organized in terms of the structure provided by **MAMBO**. Let us first consider a simulation workflow for the evaluation of the chemico-physical properties of a molecular aggregate made of identical molecules by molecular dynamics (MD). While simple, this workflow exhibits the main features of more complex simulations. The consistent representation of this workflow within **MAMBO** can therefore be instructive of the approach pursued and gives possible hints of the ability to formalize more complex cases. This macro-task can be decomposed into several interconnected computational sub-tasks, which involve different operations on structured data. From the practical point of view, the overall workflow is generally realized by applying specialized simulation software, which implements specific computational methods, operating on structured input files and producing output files as results. Other operations may require the manipulation of files and data structures. In the case of the workflow considered, we need for example input files containing information about the structure of the molecule considered. This information is further processed by specialized software, implementing computational methods, which provide an output in terms of molecular properties. The methods considered can include for example structure manipulation tools (simulation box builders, etc.) and MD specific algorithms for equilibrating molecular aggregates in different conditions[97, 10]. The workflow produces structured information containing for example a snapshot of the structure of the simulated aggregate in the conditions considered and/or derived properties (for example, the computed equilibrium density of the aggregate in $kg\ m^{-3}$. A sketch of this workflow is shown in Fig. 3.5.

An example of the parallelism between the structural information on a molecule stored as a file and encoded in a standard format in the context of molecular simulations (xyz format) and corresponding attributes of **MAMBO** classes is shown in Fig. 3.6. A similar example for attributes of classes pertaining to the `ComputationalMethod` class is shown in Fig. 3.7.

The link between the structure provided by **MAMBO** and the data defining a specific computational workflow can be provided by metadata and/or

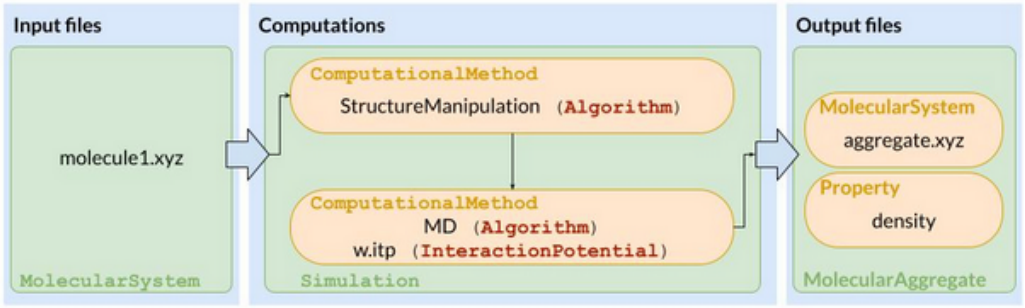


Figure 3.5: A visual description of the workflow discussed above. The first block contains the input files, which are representable as `MolecularSystem` instances (see also Fig. 3.6) as individuals, while together they are an instance of `MolecularAggregate`; the second block consists of all the files and software needed to perform the actual simulation (see 3.7 for more details); the third block represents the output obtained from the simulation, with information about the structure of the molecular aggregate and the resulting computed density.[132]

annotations, which can be implemented in a variety of standard formats[102]. The applicability of **MAMBO** in the definition of the workflow considered and analyzed by applying PSM techniques (competences - input/output, operational specifications and requirements) shows the potential of the approach proposed in the context of specific applications in the materials development pipeline. Moreover, the relatively simple case considered, in terms of concepts and knowledge considered, can be easily extended to more complex systems and processes. The semantic interoperability ground provided by **MAMBO** in the domain considered is at the basis of the representation of complex workflows in terms of basic and reusable building blocks and enables high-throughput and automated data processing. Moreover, logically defining the various steps required to link a scientific question to its corresponding results, this enables the possibility to easily re-implement workflows using different softwares or methods, while enabling interoperability between

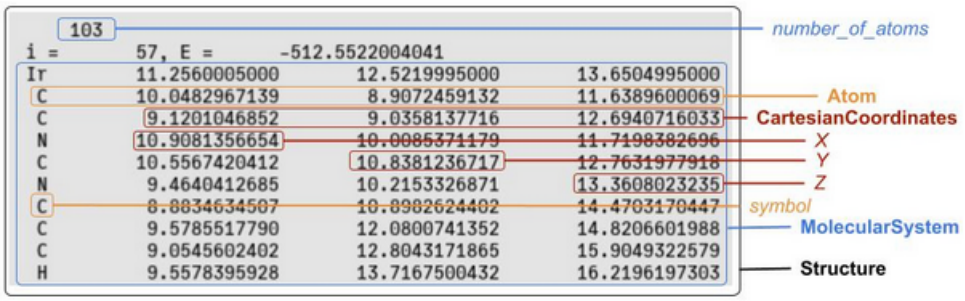


Figure 3.6: An excerpt of a real-world input file containing structural information about a molecule encoded in the standard xyz format. In particular, the file contains information on the cartesian coordinates and symbols of all the atoms in the molecule and the total number of atoms. Some of the involved **MAMBO** instances and class attributes are highlighted in different colors. Black: `Structure` instance; blue: `MolecularSystem` instance; yellow: `Atom` instance and attributes; red: `CartesianCoordinates` instance and attributes.[132]

different stacks.

3.3.1 Uniform standards for data exchange and reuse

One specific example of using **MAMBO** to aid research is the conversion to and from the various data-formats used to describe chemical entities to a unified format, which is based on the semantic structures defined inside **MAMBO**. Converting different formats to a common reference format is the basic requirement for the establishment of shared platforms for data storage and sharing, and preserving the ability to re-convert the standard format to the existing ones is fundamental to preserve the ability to use existing softwares and routines.
First, we define some conventions:

- Files containing data about chemical entities should be considered instances of the class `Structure`.

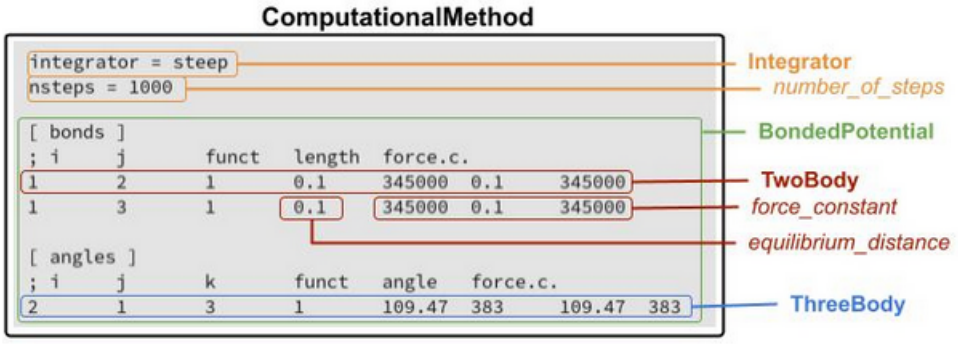


Figure 3.7: An excerpt of a real-world input file containing structural information about a molecule encoded in the standard xyz format. In particular, the file contains information on the cartesian coordinates and symbols of all the atoms in the molecule and the total number of atoms. Some of the involved **MAMBO** instances and class attributes are highlighted in different colors. Black: `Structure` instance; blue: `MolecularSystem` instance; yellow: `Atom` instance and attributes; red: `CartesianCoordinates` instance and attributes.[132]

- Inside those files, we are going to insert the information about the different `StructuralEntity`-ies that compose that specific `Structure`.
- Each of these files contains a flag called *main_class*, which tells us what kind" of structure we are going to analyze. So this flag could be one of `MolecularSystem`, `StructuralUnit`, `Particle` or `Atom`.
- In any case, the file will contain the values of the specific properties of that `StructuralEntity` (for example, a file containing the data of a `MolecularSystem` is going to have a value for `number_of_atoms`) if known.
- The same file will also contain the information regarding all the other `StructuralEntity`-ies that have a `is_part_of` relation with it; in the same way, the file contains all the needed `TopologicalEntity`-ies.

For example, a file containing the data about a `MolecularSystem` is also going to contain the data about all the `Atom`s that build that `MolecularSystem`. These sub-entities are also going to "contain"[3] all their specific information (for example, one these `Atoms` is going to contain all the information about the other `Atoms` to which is linked in some way)

For the actual formats, we chose to use the **JSON**[102] file format. This choice has been made mainly because **JSON**, other than being an open format, is a well established technology, and is the de-facto standard for data exchange (especially on the web). This has several benefits:

1. Nearly all languages already have an efficient parser for **JSON** files, together with an easy way to create them
2. While being efficiently parsed and generated, it retains a good amount of human-readability
3. There are already many databases[4] and data-driven applications[5] that are using them to store data

However, **JSON** also has some drawbacks, mainly the lack of support for dates and binary data and the fact that **JSON** objects do not have fixed length, which causes performance issues. To overcome these limitations, we could use the **BSON** format (which stands for Binary **JSON**). Other than having more built-in types, it offers fixed size objects; more precisely, under the hood a **BSON** is actually a perfect copy of a **JSON** file in binary format. So, we can adapt our workflow to use **JSON** as a general rule and then use **BSON** when performance is critical or when we need the types present in **BSON** that **JSON** lacks. **BSON** does have an obvious drawback:

[3]By "contain" we mean an additional level of nesting in the JSON file

[4]Like **MongoDB** and Apache **CouchDB**. Even **MySQL** now have support for storage and queries on **JSON** files, and many ML and DL libraries are developed in order to make **JSON**s easily usable as input.

[5]A particularly relevant example is **Elasticsearch**

being a binary file, it is not human readable, making it a good choice as an operational file format, but not for storage and human interaction. First of all, to convert the existing files to our custom **JSON** file, we must be able to read those files. For this purpose, we chose to use the excellent *Chemfiles* library[6]. We chose *Chemfiles* for two main reasons:

1. Being developed to be used as a tool for aiding computational simulations, it is very focused on high performance

2. It has bindings for many different programming languages[7], making it easier for us to develop tools that can be easily used and adapted by many different research teams

Chemfiles gives us an easy way to read all the most common file formats used to represent chemical entities.
Once read, we need to write the needed information inside the new **JSON** file. First of all, we need to gather all the data properties defined in **MAMBO** for the specific entity at hand. If these data are present in the original file, we save them in the **JSON** with the data property name as the key associated with the value. We then loop over the sub-entities contained inside the file[8] and insert a **JSON** table for each of them, and inside these tables we insert their data properties, the respective Coordinates instance and so on.
In figure A.1 and A.2 (see Appendix A, we present an excerpt of the original files (namely, an .xyz file and a .pdb file) and the corresponding resulting **JSON** file.

[6]https://github.com/chemfiles/chemfiles

[7]In particular, there are official bindings for Python 2 and 3, Fortran, C and C++, Julia and Rust.

[8]These are the ones that have a relation of type `is_part_of` with the main entity represented by the whole file

3.4 Discussion

In this chapter, we focused on the need for strong data organization policy in the field of materials science. We tackled the problem starting from the needs and hurdles faced in our first work, introduced in chapter 2. In particular, we talked about how semantic technologies and ontologies in particular can help overcome these limitations.

We've gone through the whole development process that led to **MAMBO**, the **M**aterials **A**nd **M**olecules **B**asic **O**ntology. In **MAMBO**, we have thoroughly organized the hierarchies of concepts, physical entities and properties needed to communicate knowledge and results in the realm of materials science. We gave particular attention to the computational realm, trying to fill some of the gaps we found in literature regarding molecular materials, which is our specific sector of expertise. Our development has been grounded on previous works, addressing the specific requirements of our day-to-day activities there were missing in those ontologies. For the experimental part, we directly linked **MAMBO** to a pre-existing ontology, **CHMO**, which is a very mature and extended work for experimental knowledge representation. The work has been done following the guidance of field experts, gathered through meetings and questionnaires.

We also performed representation experiments using **MAMBO** classes and relation to see if we could use them to formally represent our activities and computational experiments. If this holds true, we can use **MAMBO** axioms to represent the knowledge of our field and use these representations to build PSM, which are formal statements of problems which can then be tackled with the appropriate techniques (which are or can ben also based on the PSM formulation); moreover, we are able to tackle the data-formats problem: using the classes defined inside **MAMBO** and their mutual relation, we can develop standard encodings for representing the different entities[9] that come into play during computational simulations and related activities. With these

[9]These can be the actual chemical/physical entities or event the specific computational methods and the corresponding parameters

standards, we can then develop specific parsers to convert the standardized format to and from every specific format needed by the numerous software used to perform the actual simulations. This way, we are able to enforce those very optimized softwares to perform the heavy computations, but the results, methods and recipes can then be saved and shared in a unique way, ready to be used by different teams using different softwares stack. While this process can be arbitrarily complex, depending on the specific software and files used, the intrinsic complexity of a simulation and other specific requirements, here we provided a proof of concept for some structure files and some simulation recipes and the same reasoning and process can be extended to the more complex scenarios.
Another key aspect is that the already discussed complexity of this realm makes it impossible to develop relational databases that are suited to manage the data stemming from actual research activities. Moreover, it is also very hard to use existing **NoSQL** databases without ending up limiting the expressive power of the resulting architecture and software. For this reason, many teams are working on developing solid solutions for organizing data lakes[171] without the common drawbacks of this kind of (potentially) completely unorganized collection of data; data lakes store completely raw data, usually in the form of files or blobs. In this context, ontologies can serve as a key element of organization, allowing for the development of semantic-based engines that can enforce all the intrinsic structure of the information at hand without adding artificial constraints and limitations that are proper to existing, unspecific solutions. Technologies like **SPARQL** are the results of this kind of endeavor.

Another key prospect is that, by merging all these semantic technologies (ontologies and the consequently defined file formats, PSM, **SPARQL** and semantic queries in general), we can progressively automate different classes of problems and workflows. In the long run, and merged with learning-based techniques, this mechanism can end up becoming a new kind of expert sys-

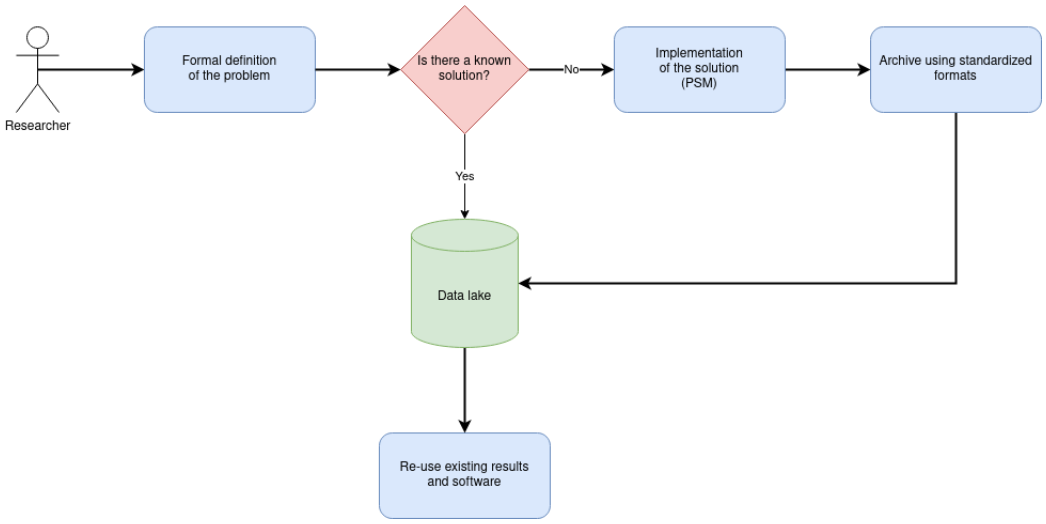


Figure 3.8: Visual representation of a semantically structured research workflow. A researcher with a scientific question can formulate it using the formal definition given by the ontology (in our case, **MAMBO**), then a semantic engine can query the data lake containing all the known results and solutions. If one already exist for the problem at hand, it can be reused; otherwise, the researcher can implement it using a PSM (based on the concepts and relationships offered by the reference ontologies), and then the resulting files are saved in the data lake using the standardized formats.

tem, able to automatically create the computational workflow corresponding to a specific scientific question. This kind of semanticallly structured workflow is represented in figure 3.8.

At the moment, **MAMBO** is only suited for research activities that deal with materials on their own, and does not define the concepts and relations needed to represent knowledge about full devices or, more generally, the interaction that takes place at the interface between two materials bulks. In the next chapter, we are going to focus on some work we made on this very field, and the struggle and limitations we are going to find are then

going to be the base for a future extension of **MAMBO**[10] to devices or for the development of a new ontology, specific for devices, that is going to use **MAMBO** as its base for concepts related to the individual materials of which a device is built.

[10] And all the related work

Chapter 4

Extracting device properties using machine learning and simulation results

The obvious next step after learning molecular properties and features is to move to the other side of the multiscale spectrum: full devices. They are very different from our previous test case on molecules and atoms, but they share the same level of complexity, which is due to the interaction between different components and higher-scale physical phenomena instead of being the cause of the interaction between many atoms and their quantistic properties.
In particular, we focused on a class of devices which are pretty fundamental for modern technologies, namely the transistors and in particular Light Emitting Transistors (LET) and their organic variants (OLET).
In this work, rather than studying the properties of different materials used in this class of devices, we focus on applying machine learning techniques to extract properties from these devices, both quicker than a standard computational simulation and using easily measurable properties to infer more "hidden" ones. Eventually, the aim of this work is to use data coming from computational workflows in order to help experimental researchers to bet-

ter understand and evaluate the quality of their devices by giving them the value of selected properties of a real-world device that would be otherwise very hard or impossible to directly measure.
To do this, we first and foremost need to choose a simulation software to use in order to obtain our data and then fit these simulated results to experimental ones. We chose to use **OghmaNano**[143, 103, 104], a relatively new tool developed at the university of Durham. We chose it both for its flexibility and its performances, paired with a good set of tools for partially automating both data collection and preprocessing. On these bases, we then developed new tools and implemented machine learning and deep learning techniques to extract the properties discussed in the following sections. Other than using machine learning techniques to solve the specific problem at hand, we also worked on improving the level of automation of the entire pipeline. As already stated, **OghmaNano** already give some instruments to automate part of the process (namely: the parameter scanning and the storage of the desired, simulated informations in a single dataset), but the whole data processing and data analysis and the actual ML pipeline of the workflow are still completely manual and left to the single researcher. Here, we furtherly automated this second part of the process, and the result is a two-step process: perform the simulations and then run the whole data-driven pipeline. This is represented in figure 4.1.

While **OghmaNano** improves and progressively offer more API to interact with it even without a GUI, we plan to completeley automate this process. The ultimate goal is to adhere to the framework that we highlighted at the end of chapter 3, an in particular in figure 3.8. Once optimized, this could be a first example of a pre-existing, reusable solution (namely: the development of a ML model able to predict device properties).

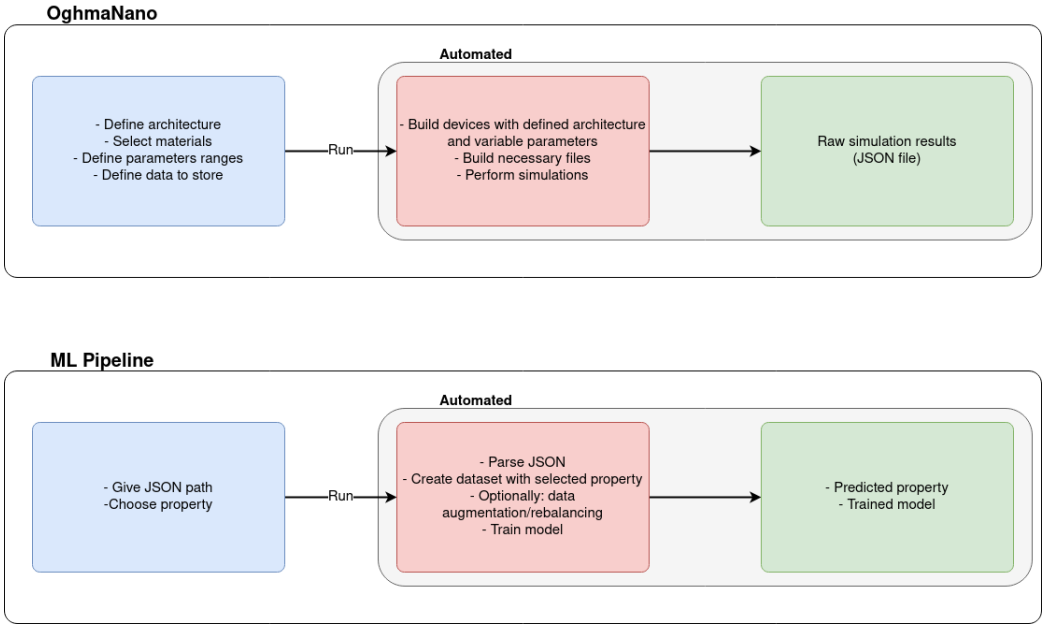


Figure 4.1: Visual representation of the two components that build the pipeline that goes from the definition of the scientific question about transistors to the fitted ML model and the predicted results. The two leftmost block (in blue) contains the manual actions that the researcher has to do personally. The other blocks (in red and green) describe all the operations that are already automated.

4.1 General context

In chapter 1, we introduced the general knowledge regarding transistors as a family of devices. Here, we introduce the differences that arise when the semiconductor of the transistor is a molecular organic material, which is the scenario where we actually worked on.
While in traditional transistors the semiconductor material used is typically silicon, in the last years there has been increasing interest in the use of organic materials as semiconductors. There are many reasons for this:

- Potentially low-cost and scalable production
- Ability to tailor their properties through the design of the molecular materials.

As already stated (chapter 1), the physics of current flow in transistors built with organic materials as semiconductors is somewhat different from that in transistors built with silicon-based materials. Here is a quick recap of the main differences:

- In silicon-based materials, the electrons can move freely through the material, giving high conductivity. In organic materials they tend to be blocked into specific energy levels, and this can lead to lower conductivity. Compared to that of silicon-based materials, electron mobility in organic materials is typically much lower
- The flow of current in silicon-based transistors is controlled by band-like transport. In organic transistors, the flow is controlled by hopping-like transport.

The main challenges that arise in using organic semiconductors in transistors are the same challenges researchers face for organic molecular materials in general: the lower conductivity of these materials when compared to silicon-based materials and their stability (cfr Chapter 1).

Despite these challenges, several teams are working with these materials in order to overcome their limitations and substantial progress is being made year after year, and the hope is to reach the point where these materials will be able to play a significant role in the development of next-generation electronic devices, replacing or helping classical silicon-based materials.

4.2 Defining the device architecture, components and conditions

To settle up this kind of activity, we need to be considerate about two things: the feasibility of the simulations (both in terms of accuracy and time required to perform them) and the availability of experimental data that are comparable to our computational setup in order to be able to fit them and adapt our system to the real-world scenarios.
For these reasons, we decided to look at a relatively simple transistor architecture with a gate, a source and a drain contacts all made of gold, an active layer made with an organic semiconductor and an insulator made of **PMMA**, a well-known organic dielectric material often used in organic electronics[186]. This is a very simple yet realistic architecture, built with standard materials which ensures good comparability with existing results, also allowing us to perform many simulations with ease. The device architecture is shown in Figure 4.2.

4.3 Property extraction

We planned a multi-step simulation, starting from a simple property in order to test our entire pipeline, the good functioning of our automation process and our ability to produce a sufficient amount of data. In particular, we decided to use the JV curve of the simulated devices as a feature vector in ML models. This choice is due to the fact that the JV curve is easy to measure in real-world devices. Indeed, if we are able to use JV curves to fit

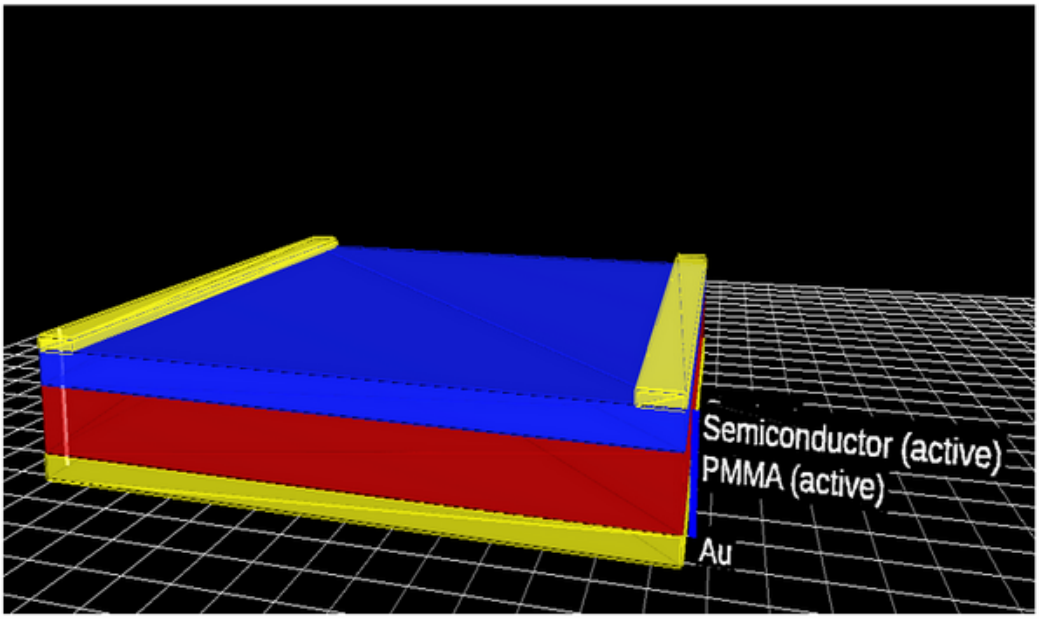


Figure 4.2: a figure showing the device architecture as rendered by the **OghmaNano** built-in CAD software.

other (and harder to extract on actual devices) properties we can help experimental researchers to better profile their devices and to extrapolate more knowledge on physical phenomena from the same experiments and devices.
First of all, we decided to predict the mobility of carriers in a device, including both the full-device isotropic value and the component in the in-plane direction (from source to drain). The mobility is generally the property that is most related to the overall current output in transistors and, consequently, to the JV curve. The fit of mobility allows us to investigate future possibilities, while validating our computational approach. Next, we simulated the behavior of devices at different temperatures, to see if the impact temperature on simulations can hinder or help the learning process.
Then, we decided to use simulations to compute the mean density of electrons and holes traps. These are crucial properties for the performance of molecular materials applications in electronics because they strongly influ-

ence the amount of current that flows through the device.
Lastly, we tried to use the same approach to predict the density of carriers in different regions of the device. Being able to do this for several parts of the device would give us the possibility to reconstruct the carriers density in the device, giving researchers a clue on the deeper structure of the device, and helping them in identifying possible performance bottlenecks, architectural issues and more. The distribution of carriers density across device sections can only be extracted from real-world devices with complex experimental set-ups, while also being quite expensive to be simulated accurately on-the-fly.

4.4 Defining the pipeline

Our computational approach is structured as a layered pipeline of procedures and includes all the standard steps needed for conducting a set of device simulations together with the steps needed for using ML techniques. In the following sections, we discuss this workflow in detail.

4.4.1 Simulations

As already mentioned, we used the **OghmaNano** software to perform simulations. **OghmaNano** is a flexible software, which also offers a very powerful GUI interface that is suited to be used by researchers that do not have prior experience in this kind of operations. **OghmaNano** is a mesh-based simulator for numerically solving drift-diffusion equations in devices. This means that device properties are mapped onto rectangular meshes. Lower-level chemical and physical properties of materials, their interfaces and the consequences on the whole device are taken into account parametrically using numerical parameters. The resulting approach is very efficient and fast, allowing us to simulate a large number of device configurations in a relatively short time and to rapidly gather the amount of data needed to train our ML models and neural networks.

4.4.2 Data management

OghmaNano offers a system to automatically perform the parameter scanning we discussed in the previous section: for each point in the parameter scanning grid, a simulation with these parameters is performed. More specifically, in **OghmaNano**, each different simulation has its own folder, where the results are saved. Then, all the results are easily gathered inside a .json file, where each different simulation corresponds to a univocal hash, to which corresponds a **JSON** Object. Inside this table, the different parameters are identified with a per-name criterion, corresponding to different **JSON** basic types, like boolean, number and array. This structure is easily parsable using any json-related library for any programming language of choice, giving the possibility to easily move from the **JSON** file to a more common programming interface for tabular data, like DataFrames.
However, there is another important data that must be considered: the configuration files needed to perform the specific simulations. By default, these are saved together with their corresponding results, and they are also saved in a **JSON** file. This file, despite its complexity, can still be parsed with any **JSON** library, allowing us to be able to reuse and modify input files when needed[1].

4.4.3 Machine learning

We decided to start with a relatively simple machine learning-based approach. We wanted to start again from what we learned during the activities reported in chapter 2, trying to replicate the same approach as much as possible. Moreover, differently from the case of molecular systems, we are able to easily and quickly generate large datasets, thus limitating the need for

[1] Actually, behind the curtain, **OghmaNano** uses a basic version of input files to generate the files needed for any simulation, altering the needed variables and parameters.

augmentation techniques like **SMOGN**.
For these experiments, our data pre-processing pipeline is roughly the same as the one highlighted in chapter 2, and also the chosen ML algorithm is the same algorithm that performed better: **XGBoost**. Also, the tuning strategy implemented is the same discussed in chapter 2
For the last, more complex experiment, we evaluated different fully-connected neural networks, since the relation between the features and the targets we wish to fit is evidently more complex and known to be not linear. Since our input data does not exhibit any temporal relation (time series), textual or generally recurrent structure nor is made of images, we rely on the simplest possible neural network models. In future experiments, it could be interesting to study the effect of giving a more powerful "a priori" knowledge to the model.

4.4.4 Automating the pipeline

To make the entire work easier and quicker, we developed a system to automate the whole simulation and analysis pipeline. **OghmaNano** offers a set of tools to automate the randomization of the parameters and generate the different simulations files, gathering them in a common output file. Then, this file is parsed by a python script that does the pre-processing step, which is then followed by the actual ML procedure. While conceptually simple, this step is fundamental for giving researchers the possibility to efficiently carrying out simulations and to enable the generation of large datasets for the application of ML techniques. Moreover, this entire procedure has been implemented using well known languages and formats, i.e. python and **JSON**. The choice of common standards and formats enables the share of procedures with a broader community and to re-use parts of these recipes to automate generic workflows. In the long run, these automation procedures can serve as the base for a more general automation engine for the whole computational materials science domain.
This automation effort has been developed keeping in mind the assumptions

and reasoning highlighted in chapter 3. Despite lacking some of the needed concepts, the semantic assets offered by **MAMBO** has been used as the basic pillars during the whole process, and this served two purposes:

- serving as another example of the applicability of **MAMBO** (and semantic modelling in general) to materials science R&D activities.
- helping us in identifying new classes and relationships that are needed in order to make **MAMBO** a complete reference ontology for materials science and devices development.

4.5 Experimental setting

We performed four different experiments, each of which led to the creation of a separate dataset. In each experiment, we set **OghmaNano** in order to perform a campaign of simulations on different devices with the same architecture while varying different parameters in order to test the ability of our model to generalize to different and more difficult scenarios. For all the different experiments, we sampled the ***JV*** curve at 10 different voltages to monitor the evolution of the system. This is going to be the feature used to train the models in order to predict the different targets in each experiment. This means that the models are going to be trained using an array of 10 real numbers for each entry of the dataset. For each experiment, we performed simulations in order to gather a dataset with 1.000 entries.
For the **first experiment**, we configured a device to be either pure electrons- or pure holes-based, in order to make the problem as simple as possible. We then performed the simulations randomly scanning the average mobility of electrons or holes in an interval ranging from $1e^{-10}$ to $1e^{-3}$ $m^2V^{-1}s^{-1}$ both for the elecrons-based devices and the holes-based devices. We then used the obtained JV curves in order to train the model to calculate the electrons/holes mobility.
We did an analogous thing for the **second experiment**, but this time, other

than average mobility, we also randomly modified the temperature, the tail slope and the traps density[2]. We then trained different models to predict the average mobility again and also the tail slope and traps density.
We then started to also alter the geometry of the devices; in particular, the **third experiment** has been divided in three different sub-experiments, where we randomly altered the thickness of the **PMMA** layer, the channel width of the device and then both parameters. Again, we trained different models in order to predict the carriers mobility and the tails slope and traps density.
Lastly, in the **fourth experiment**, we also performed simulations sampling the density of the carriers at specific horizontal slices of the semiconductor layer. We then trained our models to use the same JV curve to predict this carrier densities. This is a harder problem, and we used it in order to test the power of our approach in more difficult situations.

4.6 Results

Upon definition of the simulation conditions and of the ML and DL workflows, we then proceeded to assess our approach. The whole process was assessed iteratively, i.e. simulating a new architecture while using the previous one to feed the respective ML/DL model.
All the different results were analyzed separately and comparatively, in order to see the difference in the results depending on the complexity of the problem and to fully understand the value of our approach.

[2]Te temperature ranages from $280K$ to $320K$, the traps density ranges from $1e^{+15}$ to $1e^{+26}$ $m^{-3}eV^{-1}$ (both for electrons and holes); the tail slope ranges from $1e^{+15}$ to $1e^{+26}$ eV.

4.6.1 Fitting mobility in basic device architectures

As expected, this first experiment proved to be quite easy to solve. Even with a few hundreds of simulations we were able to fit the mobility of the majority carrier in OFETs nearly perfectly. However, the model is not able to predict the same value for minority carriers. This is an expected behavior, since in a single carrier (unipolar) device the behavior of the minority carrier does not follow a regular pattern and its flow is more due to recombination and other side effects. We also tried to use simpler models, like random forest and a ridge regressor, still achieving very good results. A plot showing main fitting results is depicted in Figures 4.3.

While simple, this experiment proved to be fundamental in order to establish that, at least to some extent, the JV curve is actually a good feature to predict properties of a device. In particular, it allows us to have a nearly perfect prediction of the mobility of the major carrier inside a device.

4.6.2 Fitting mobility with variable temperature

In this second experiment, we introduced a first variable to test the effect of a slightly more complex system on the fit. From our tests, the varying temperature does not visibly affect the quality of the fit, as visible in figure 4.4.

It must be noted that we did not give the temperature to the model as a feature. This could be an indication of the quality of the JV curve as a feature to predict properties of devices. This assumption in further tested in the experiments described below.

4.6.3 Fitting mobility with variable thickness and variable channel width

Our model managed to fit the problem very well even without using a very big dataset. Again, the JV curve proved to be a sufficient feature to predict the carrier mobility. This means that the information regarding the

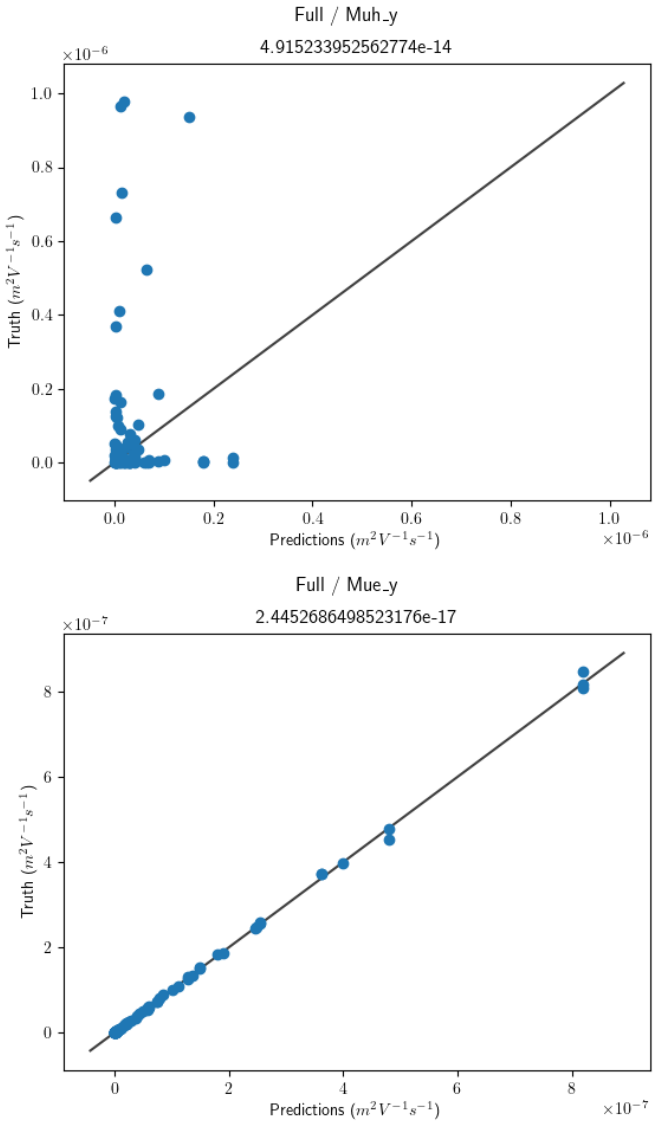


Figure 4.3: Plots showing the relationship between values of the holes (1) and electrons(2) mobility in unipolar p-type OFETs predicted from the ML algorithm (on the x axis) and those obtained from the simulations (on the y axis). To make it easier to evaluate the results, the x=y line is drawn. Similar results are obtained for p-type devices. As expected, the model fits the relation between the JV curve and the mobility of majority charges, while it cannot fit the relation between the JV curve and the minority carriers mobility.

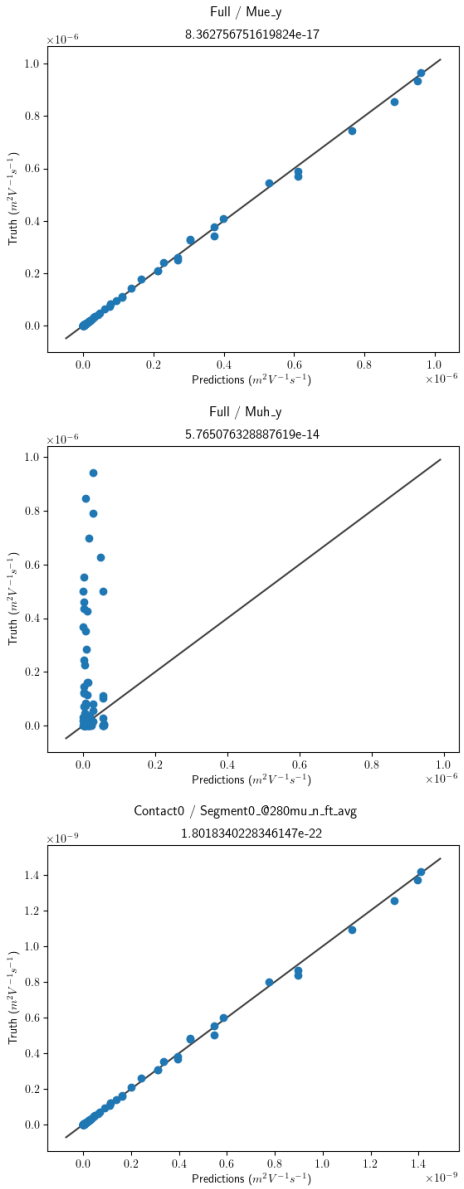


Figure 4.4: Plots showing the relationship between values of the mean electrons mobility with constant temperature (1), mean holes mobility with varying temperature(2) and directional electrons mobility with varying temperature(3) predicted from the ML algorithm (on the x axis) and those obtained from the simulations (on the y axis). To make it easier to evaluate the results, the x=y line is drawn. These data are related to simulations coming from electrons-only devices, but specular results are obtained for holes-only ones. As expected, the model can fit the relation between the JV curve and the electrons mobility, while it cannot fit the relation between the JV curve and the holes mobility. We can see how the variation in temperature does not affect the fit, which is still nearly perfect.

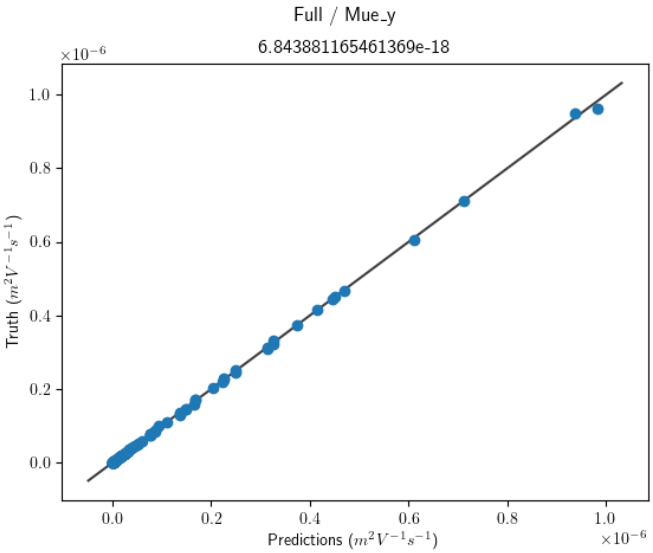


Figure 4.5: A plot showing the relationship between values of the electrons mobility with varying thickness of the semiconductor predicted from the ML algorithm (on the x axis) and those obtained from the simulations (on the y axis). To make it easier to evaluate the results, the x=y line is drawn. These data are related to simulations coming from electrons-only devices, but specular results are obtained for holes-only ones.

thickness is somehow embedded in the JV curve. The results are shown in Figure 4.5.

Experiments performed by varying the channel width led to analogous results (see Fig. 4.6). Therefore, these first experiments suggest that the JV curve is somewhat able to give to the model some indication about the geometrical characteristics of the device. This observation will be put to test in the next section.

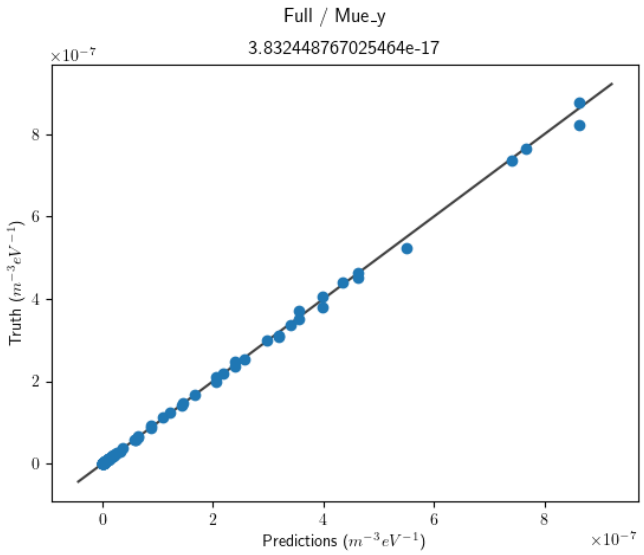


Figure 4.6: A plot showing the relationship between values of the electrons mobility with electrons mobility with varying channel width of the semiconductor predicted from the ML algorithm (on the x axis) and those obtained from the simulations (on the y axis). To make it easier to evaluate the results, the x=y line is drawn. These data are related to simulations coming from electrons-only devices, but specular results are obtained for holes-only ones.

4.6.4 Fitting mobility varying both thickness and channel width

Even varying both the thickness of the active layer and the channel length, our model is still able to correctly predict the mobility of devices from JV curves. This result further demonstrates that the JV curve also contains some information about the geometric characteristics of the device, giving us the possibility to fit different device architectures when dealing with carriers mobility. It looks like that using the JV curve as our feature we are able to capture the general structure of the device, and that the variation of some structural/geometrical characteristics (like thickness of layers and channel width) is directly correlated to specific variation of the JV curve. Moreover, good fits are obtained by considering either the mean current density in the whole active layer or the in-plane component (source-drain). In figure 4.7, we report the results of our fitting in this specific case.

4.6.5 Predicting traps density and tail slope

We then tried to extend the approach proposed to predict other properties of interest in devices, moving to a more complex setting. In particular, we considered the prediction of the traps overall density and the tail slope of electrons and holes distributions.
At difference with the cases discussed above, the approach presented in the previous sections is essentially unable to predict the trap density distributions. The results for traps density and tail slope in the same setting as 4.6.2 are visible in figure 4.8 and 4.9 respectively.

The reasons for this failure in predicting correct values can be ascribed to the limited direct relationship between the trap density and the overall performance of the device, represented by the JV curve. Moreover, the density of traps is generically associated to intrinsic material and fabrication properties. As such, the determination of the trap density by using the JV curve only is probably unfeasible without considering other materials properties.

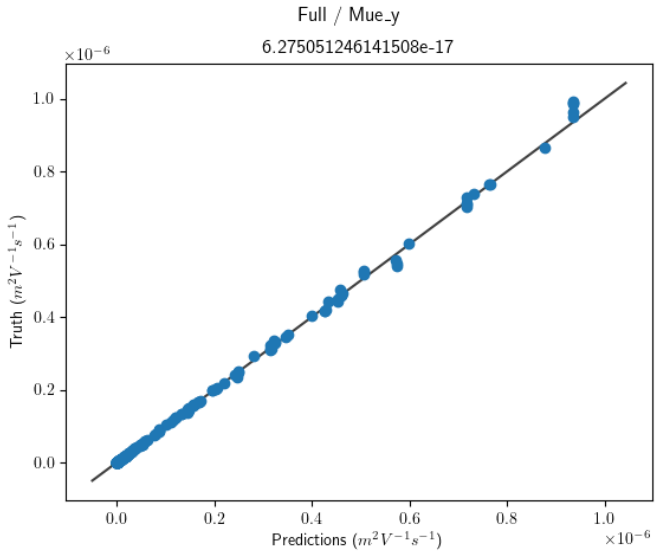


Figure 4.7: A plot showing the predicting ability of the model when both the thickness and the channel width are varied. The overall quality of the fit is still very good, with a small deviation at very high values, which can be ascribed to a statistical anomaly due to the limited number of samples when compared to those with low mobility values.

In addition, the trap density used in numerical simulations represents phenomenological parameters built upon approximations meant to make molecular materials more akin to classical semiconductors based on silicon. Accordingly, the difference in the physics behind the electronics between these two classes of materials (highlighted in section 1.1) explains why the formalism based on traps is not perfectly transferable to molecular materials, in which the main charge transport mechanism is due to hopping, with a less relevant role of energy level distribution.

Although we are able to simulate the behavior of organic electronic devices by essentially transfer the formalism used for silicon-based systems, the value of specific simulation parameters can be related to very different physical properties in the two cases. Accordingly, the ML model is not able to fit a number that is no a real quality of the device or the materials, but

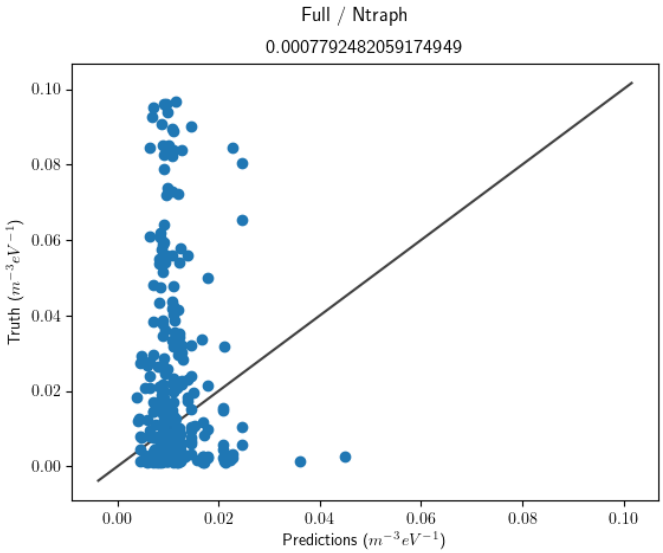


Figure 4.8: A plot showing the results of the model in predicting holes traps density. We can see how the model fails at understanding the relationships between the JV curve and the traps density, resulting in a very poor fit.

a numerical construct used to describe the charge transport mechanism of molecular materials in the same way of classical silicon-based materials. For this reason, this is a quality that is derived from progressive approximations of experimental results, fitting the simulation results to those real-world devices.

4.6.6 Predicting carrier density distributions

In this last, more complex, experiment we assess the performance of ML models for predicting the distribution of carrier densities across the active semiconductor layer. The density of charge carriers in devices is strongly correlated with the JV curve. However, the spatial distribution of carriers density is also an intrinsically geometric and material dependent property, making it akin to the traps density and the tail slope.
It turns out that the nature of the material is predominant in determining

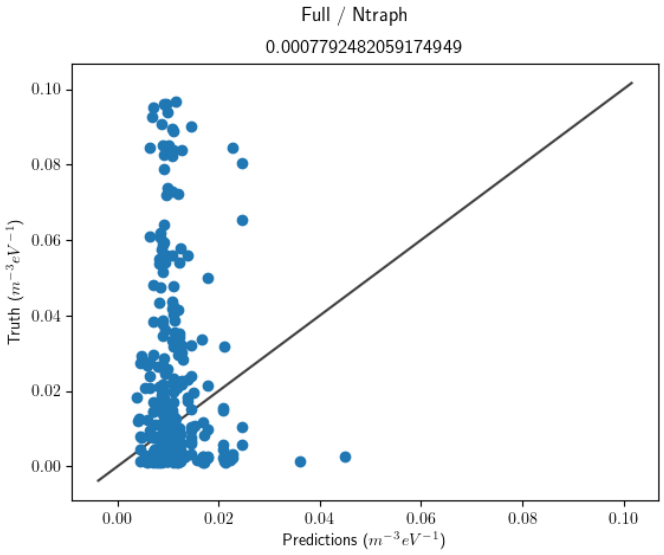


Figure 4.9: A plot showing the results of the model in predicting holes tail slope. We can see how the model fails at understanding the relationships between the JV curve and the tail slope, resulting in a very poor fit.

this property, and again we ended up failing in fitting a model to predict it. This is visible in figure 4.10.

This is another proof of the fact that, for predicting properties strongly related to the specific materials used to build the device we need specific features that describe the atomistic or molecular arrangement of at play, or at least a more accurate formalization of the properties of the materials bulk. This is another indication that ML models can play a huge role in enabling multiscale simulation and multiscale investigations in general of materials and devices: models at the molecular scale (see 2) can be used to predict materials properties, which can be plugged into device simulations or, for example, to train higher scale models for the prediction of full-device properties, as carrier or traps densities or the tail slope. On top of that, these models can be efficiently integrated with each other through the huge throughput boost enabled by the ML approach, leading to an even deeper understanding of physical models.

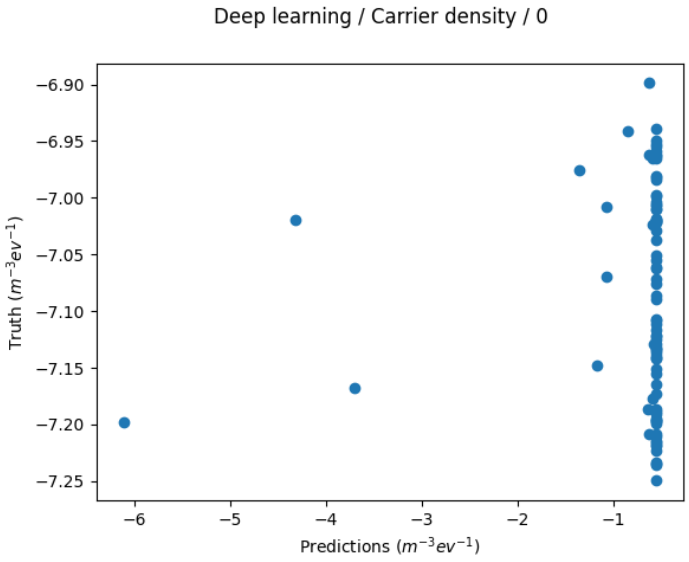


Figure 4.10: A plot showing the results of the model in predicting carriers density at the bottom-central part of the semiconducting layer. We can see how the model fails at understanding the relationships between the JV curve and the carriers density, resulting in a very poor fit.

4.7 Discussion

In this chapter, we discussed the role of data-driven techniques applied to device simulations and in particular to the prediction of device properties. We showed how, using the JV curve as the unique feature, we are able to predict the carriers mobility of a device even with varying architectural parameters like the semiconductor thickness and the channel width. ML models are also able to fit device properties with one of more varying parameters and physical properties, such as temperature. While showing how the JV curve also brings some geometrical and physical information about the device, we also showed how this feature is insufficient when the target is more strictly related to the materials used to build the device, in particular the organic ones. As stated in chapter 1 and as also proved in chapter 2, the structure and properties of organic materials are profoundly different from those of crys-

talline and silicon-based materials, which can be quite efficiently described in simulations using bulk-like phenomenological parameters. Consequently, the impact of materials properties on device performances is particularly hard to predict in molecular materials.

However, the inability to predict properties related to the intrinsic structure of active materials is another indication of the strong need of computationally efficient tools to calculate materials properties across a broad range of spatial scales. In addition, multi-scale crosslinks require tools that are able to aggregate knowledge, leading to effective materials parameters and descriptors. Enabling this step can lead to a whole new paradigm of multiscale simulation and multiscale scientific computing in general, allowing researchers to develop software that can simulate the performance of full devices using actual properties of the specific materials used, calculated on-the-fly through data-driven algorithms in a real-time fashion. We want to develop methods that can simultaneously exploit information coming from different scales, with the same precision and without having dividing the computation into separate tasks. This next-level integration will lead to more precise assessment of the behavior of real-world devices, additionally increasing the effectiveness of computational experiments.

Moreover, this experiment helped us in further assess the effectiveness of our semantic modelling tools and in particular that of **MAMBO**. While the general structure proved to be effective, we also highlighted some missing concepts like that of *interface*. This leaves us with a choice, namely extending **MAMBO** in order to explicitly offer new assets to represent devices and their properties, or creating a new ontology for devices that will leverage **MAMBO** for describing the individual materials that compose the devices. Since **MAMBO** has been developed as a *lightweight ontology*, we tend more to the latter option, but in order to make the best choice we need more experimentation and careful evaluation.

Chapter 5

Suggesting parameters for device optimization via ML-driven analysis of experimental data

To this point, we have talked a lot about computational workflows and resulting data, showing how we can use data resulting from this kind of simulated experiment to train ML algorithms able to predict the same properties but in a more efficient way. However, there is a huge elephant in the room: research in the materials science domain is obviously strongly dependent on actual empirical data, collected within experimental activities conducted in the lab. Other than the obvious difference between the two realms, experimental data pose even more challenges; these are due to possible difficulties in reproducing experiments, the intrinsic limited precision of the measurements, the variability of the results due to external conditions and unpredictable events (from errors of the operator, wear of the tools, to variable meteorological conditions affecting experimental processes and so on) and also due to a minor diffusion of all the aspect related to the data culture, like developing sensible formats, collecting data in a standardized way, col-

lecting as much information as possible (instead of just recording those that are needed immediately) etcetera.
In this chapter, we discuss the work done in order to overcome many of these limitations: as an application example, we considered a recent dataset generated by collecting available data on hybrid photovoltaic solar cells[64] and reporting all the known information in a single database. This procedure is obviously prone to errors, leading to incomplete and very diverse resulting information for each entry. Solving these problems is still an open challenge in the field, but in this chapter we show how data science techniques can be applied to process this (and other) available dataset to gather information about possible unknown phenomenon related to the performance of the target photovoltaic devices.

5.1 General Context

A general introduction on the principles behind the working mechanisms in organic and hybrid solar cells can be found in chapter 1. In this application example, we deal with photovoltaic cells based on a class of materials called perovskites. Here, we introduce both the main properties of perovskite materials and of hybrid (organic/inorganic) photovoltaic cells. Then, we explain the peculiarities of photovoltaic cells based on perovskites.

5.1.1 Perovskites

Perovskite materials are a class of compounds that have a specific crystal structure known as the perovskite structure. They are named after the Russian mineralogist Lev Perovski, who first observed and described this structure in a mineral (the calcium titanium oxide, whose formula is $CaTiO_3$), which is called perovskite itself. In literature, this crystalline structure is referred to as ABX_3, which represents the general chemical formula for perovskite compounds, where A and B are cations and X is the bonding anion (Figure 5.1). Within solar cells, lead (Pb) is often the dominant metal used

in perovskites.

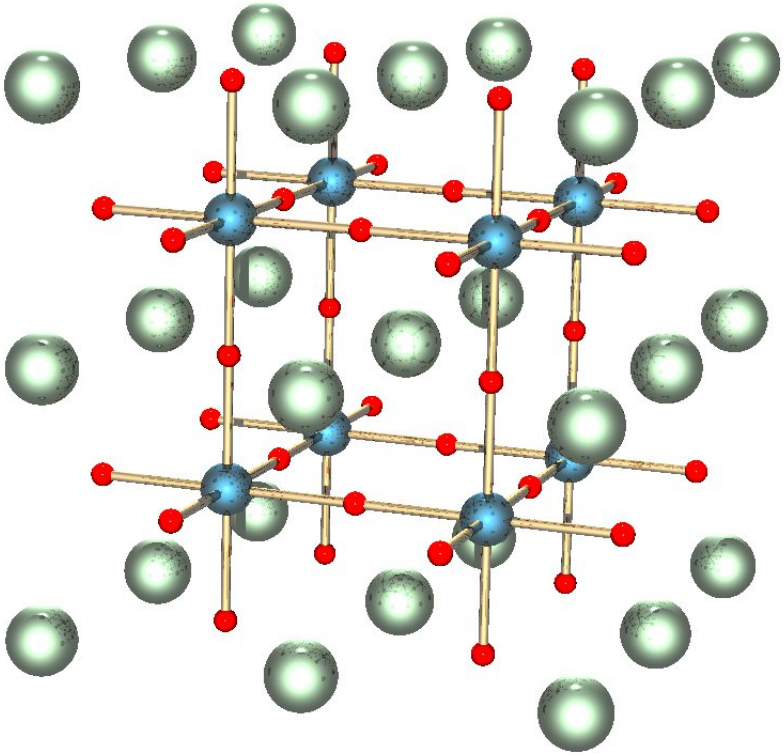


Figure 5.1: A figure showing the crystalline structure known as perovskite structure. The main element are the cubic-like grid and then the single atoms (which, in the case of electronic applications, are going to be the cations) around and inside that grid.

Perovskite materials can be used as electronic components due to their semiconducting properties. In particular, they have shown promise as active layers in photovoltaic cells, where they can absorb light and generate an electric current. They have also been explored as active layers in transistors and light-emitting diodes, where they can be used to control the flow of electrical current. In energy storage, perovskites have been explored as cathode materials for lithium-ion batteries and supercapacitors. In photovoltaics, perovskite materials have shown great potential as a replacement

for silicon used in traditional solar cells.

5.1.2 Perovskites and molecular materials

One area of particular interest for perovskites is the use of molecular materials as cations. Cations are positively charged ions that are incorporated into the crystal structure of perovskite materials. Molecular materials have several advantages over traditional inorganic cations, including the ability to tailor their properties through the design of the molecules, and the potential for low-cost, scalable production.

5.1.3 Hybrid organic-inorganic perovskite solar cells

Hybrid organic-inorganic perovskite solar cells (HOIPs) are a type of solar cell which active layer[1] is made of a material exhibiting the perovskite structure[106]. The laboratory-scale efficiency of these devices have been steadily increasing from 3.8% in 2009[80] to 25.7% in 2021.
The main advantages of this family of devices are related to those of the perovskite against silicon, like:

- As already stated, they are very efficient. Research has shown that HOIPs can have efficiencies of over 20%, which is comparable to traditional silicon-based solar cells[117]. This high efficiency is due to the unique electronic properties of the perovskite material, which allows for efficient charge separation and transport.

- The materials used and the possible fabrication methods[2] are both low cost[140].

- Traditional silicon cells require expensive, multi-step processes, conducted at high temperatures ($> 1000°C$) under high vacuum in special cleanroom facilities[144]. Meanwhile, the hybrid organic-inorganic

[1] I.e., the layer responsible for the light-harvesting.
[2] For example, the numerous printing techniques currently available (cfr chapter 1).

perovskite material can be manufactured with simpler wet chemistry techniques in a traditional lab environment. Most notably, some of them have been created using the aforementioned deposition techniques, all of which have the potential to be scaled up with relative ease[3][148, 61, 157, 73, 89].

- Their high absorption coefficient enables ultrathin films of around 500 nm to absorb the complete visible solar spectrum[179].

- Other than their simplicity of processing, perovskite solar cells hold an advantage over traditional silicon solar cells in their tolerance to internal defects[74].

- Perovskite cells also possess many optoelectrical properties that benefit their use in solar cells. For example, the exciton binding energy is small. This allows electron holes and electrons to be easily separated upon the absorption of a Photon. Moreover, the long diffusion distance of the charge carrier and the high diffusivity - the rate of diffusion - allow the charge carriers to travel long distances within the perovskite solar cell, which improves the chance of it to be absorbed and converted to power.

- Perovskite cells are characterized by wide absorption ranges and high absorption coefficients, which further increase the power efficiency of the solar cell by increasing the range of photon energies that are absorbed[25].

- HOIPs also have the advantage of being lightweight and flexible, which makes them suitable for use in portable and flexible devices[95]. This makes them ideal for use in a wide range of applications, such as in portable electronic devices, wearable technology, and flexible solar panels.

[3]With the exception of spin coating.

- Additionally, the use of organic cations in the perovskite material allows for more flexibility in terms of material choice, enabling the incorporation of a wide range of inorganic and organic materials[67], and for the development of new materials with improved properties

- Specific to perovskites using organic materials as cations, there is evidence of resulting in more stable materials and devices. This is crucial for commercialization and practical applications. The stability of the perovskite material is sensitive to moisture and heat, but research has shown that the use of organic cations in the perovskite material can improve the stability of the solar cells[87]. This is a major advantage over traditional inorganic perovskite solar cells, which are less stable and more susceptible to degradation under certain environmental conditions.

These combined features result in the ability to fabricate low cost, high efficiency, thin, lightweight, and flexible solar modules with relative ease. For example, perovskite solar cells have found use in powering prototypes of low-power wireless electronics for ambient-powered Internet of Things applications[75], and may help mitigate climate change[124].

Limitations

Toxicity This is mainly due to the lead content in perovskites[7]. While traditional silicon-based solar cells are thermally and chemically stable, perovskites are very unstable and easily degrade to rather soluble compounds of lead or tin, significantly increasing their potential bioavailability[68] and hazard for human health[4][14, 8].
Toxicity is also related to recyclability: currently, producing 1 GW of energy

[4]While the lethal dose is known to be 5 mg per kg of body mass, serious health effects arise even at way smaller doses. In particular, younger children are more susceptible to the toxic effects of lead, and it is known that lead exposure can result in decreased intelligence and behavioral problems[126]

using the most efficient perovskite solar cell would result in 3.5 tons of lead waste. It is then crucial to develop ways to reduce lead contamination with ways other than lead-leakage prevention.
However, there are some pieces of literature[41, 2] addressing many of the causes of these two huge limitations. More generally, the use of organic materials as cations for the perovskites will help to reduce the environmental impact of these devices.

Stability One big challenge for perovskite solar cells (PSCs) is the aspect of short-term and long-term stability[44]. The traditional silicon-wafer solar cell in a power plant can last between 20 and 25 years, setting that timeframe as the standard for solar cell stability. PSCs have great difficulty lasting that long[139]. The instability of PSCs is mainly related to environmental influence (moisture and oxygen)[19, 76], thermal stress and intrinsic stability of methylammonium-based perovskite[68, 69, 71], and formamidinium-based perovskite[70], heating under applied voltage[182], photo influence (ultraviolet light)[108] (visible light)[69] and mechanical fragility[145]. The water-solubility of the organic constituent of the absorber material make devices highly prone to rapid degradation in moist environments[47]. The degradation which is caused by moisture can be reduced by optimizing the constituent materials, the architecture of the cell, the interfaces and the environment conditions during the fabrication steps[108].

Hysteretic current-voltage behavior Another major challenge for perovskite solar cells is the observation that current-voltage scans yield ambiguous efficiency values[158, 167]. The power conversion efficiency of a solar cell is usually determined by characterizing its current-voltage (IV) behavior under simulated solar illumination. In contrast to other solar cells, however, it has been observed that the IV-curves of perovskite solar cells show a hysteretic behavior: depending on scanning conditions (such as scan direction, scan speed, light soaking, biasing) there is a discrepancy between the scan from forward-bias to short-circuit and the scan from short-circuit to forward

bias[158].

5.1.4 Production

This is quite similar to the process highlighted in Chapter 1. First of all, the perovskite must be synthesized, which usually means collecting the precursor materials (i.e. the individual elements and molecules that are going to be used to build the perovskite), which are then mixed together in a solvent and then heated to high temperatures (usually between 150 and 300°C) in order to form the actual crystalline structure of the perovskite. The ratio of the different materials plays a fundamental role in the properties of the final product. After that, the perovskite must be deposited on the substrate (which usually is a layer of titanium dioxide on top of a glass surface). This is done using some of the same deposition techniques introduced in chapter 4, like spin coating[20]. It is then the moment to form the actual p-n junction, and this is usually done by adding a small amount of a p-type material to the perovskite, then heating again to allow for the formation of the actual p-n junction. When using perovskites with organic cation, there is an additional final step that is the encapsulation of the solar cell in order to protect it from the environment. This is typically done by sealing the solar cell in a protective layer, such as a layer of glass or plastic.

5.2 Identifying best parameters and parameter values via machine learning

In this work, we are going to use unsupervised learning techniques to identify features that may lead to better performing solar cells. As already stated in the previous sections, the relationship between the architectural and chemical parameters of the cell and the final results are only partially known, and we want to see if some of these hidden relations can be extrapolated from the known data. The main idea is simple: use some of the known features

of many different cells, try to divide a dataset in different clusters and see if these clusters actually exhibit different performance characteristics.
Since the clustering is done without taking the performance as an input, the algorithm is not going to actually learn anything about the performance, but will make its decisions using only the features. If the resulting clusters show different performance measures, we have a strong indication that the features used are actually important in determining the properties of the final device.
The choice of performing clustering instead of simply train a model to perform regression is due to two factors:

- Since we don't want to simply get the values of the different performance measures but we want to identify the features that are most crucial in determining the performance itself (and, obviously, the values of these features that lead to better results), simply performing a regression would give us less information about these hidden patterns; performing clustering, instead, lead us to a subdivision of the dataset in different groups of devices with, hopefully, uniform performances and a mixture of uniform and non-uniform features values. If this proves to be true, we have strong indication about which features are actually important in determining the performance (i.e. the features with uniform values inside the clusters) and those that, instead, are less crucial (i.e. those with non-uniform values inside the clusters).

- As already stated, the intrinsic complexity of the discipline (namely: the fabrication of devices based on molecular and functional materials in general) means that perfect reproducibility of the results is nearly impossibile, since the result strongly depend on chaotic and uncontrollable factors like weather; this means that trying to perfectly predict the already known values of the performance measures using only the known features can lead to a model completely unable to generalize to new data and to deal with the inevitable uncertainty of the numerical results. Using clustering, instead, we can deal with this kind of uncertainty by not trying to predict the exact numerical values but trying

to predict *ranges* of values and families of devices, which can then be studied more deeply by the experimental researchers while trying to minimize or compensate for the complexity of the fabrication and measurement process.

5.2.1 The Perovskite Database

This is an important effort in creating a common data platform that researchers can head to when dealing with perovskite-based solar cells. Other than the data itself, the site[5] of the project host plotting and general analysis tools that researchers can manually operate directly from the site itself. Quoting its homepage:

> "The Perovskite Database Project aims at making all perovskite device data, both past and future, available in a form adherent to the **FAIR** data principles, i.e. findable, accessible, interoperable, and reusable.
> In the initial phase of the project, the project team went through the over 16000 perovskite papers published until the end of February 2020 and extracted data for every single adequately described perovskite solar cell we could find. For papers published after that, the database relies on authors to upload their own data.
> The project is based around an open database and open-sourced tools enabling anyone, without any programming experience, to interactively explore, search, filter, analyze, and visualize the data. The core of those tools are a set of interactive graphics that can be reached from the web page."

The general structure of the project is represented in figure 5.2.

The relevance of this project resides in the fact that (as stated in previous chapters) there are not many open big databases of experimental data available today. Moreover, the development of photovoltaic cells is a crucial

[5] https://perovskitedatabase.com/

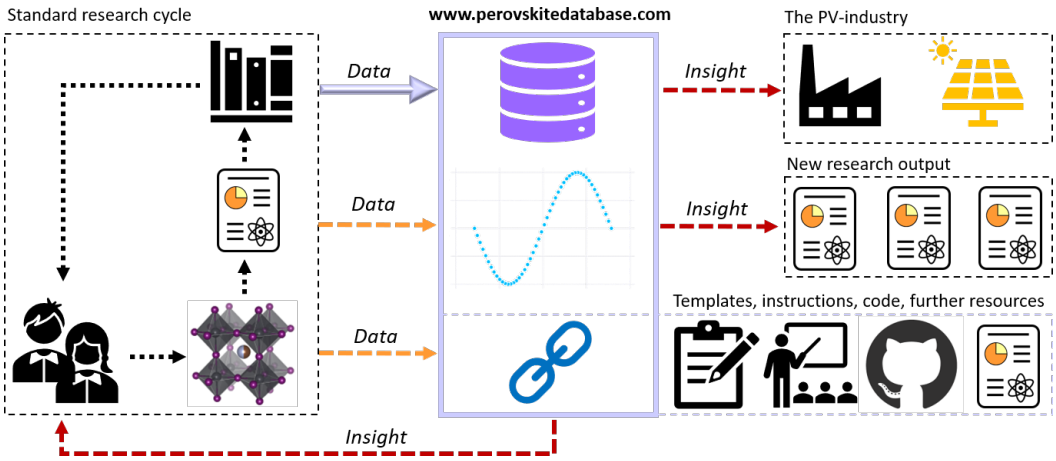


Figure 5.2: A schematic overview of the idea behind the Perovskite database. This shows the standard research cycle on the left side, and on the right it shows how the project aims to expand it, helping researchers to improve their throughput[64].

problem which suffers from all the design difficulties and problems highlighted in the introduction of this chapter and also in the previous one. For these reasons, it is extremely important to gather all the information known at this time and make them available for researchers around the world, enabling them to enforce data-centric techniques (like ML) to use such information to acquire new knowledge on the topic and developing new architectures and fabrication procedures for this crucial class of devices.
The collection procedure[64] has been quite simple, yet time consuming: authors manually gathered all the articles present in literature about perovskite solar cells, extrapolated the data contained in each paper and put them in a table-shaped database. Obviously, this process shows several challenges and limitations:

- First of all, this manual process is highly susceptible to human errors. Along time, many corrections have been implemented and highlighted on the site, but it is impossible to be completely sure that we are using correct data

- Related to the first point, there is also the possibility that the data reported in the articles are not correct (authors may have reported them incorrectly or with approximations, the publisher may have formatted them in a bad way, and so on)

- Not all articles provides all the parameters present in the database; quite the opposite, the majority of the entries are not complete

- Not all articles supply the same information in the same way, with the same units and with the same level of detail (for example, error margins)

Other than these methodological aspects, there are also other issues, related to the lack of standards and expressive formats in the field:

- Some of the features present in the dataset are meant to represent different aspects of the fabrication process. These are intrinsically complex, entries are extremely variable[6] and it has to describe an idea of causality or other time-based relations, resulting in a complex feature which is hard to use in this unprocessed form

- Many different strings are used to identify missing data, making it very difficult to find them all. Some of them are also too similar to a possible entry[7]

- Combining the two previous problems, when in a sequence of steps some of these steps are unknown, the authors inserted one of the aforementioned strings, which is mixed inside actual known data. Other than making it even more difficult to find all the missing data in an

[6]For example, the fabrication process can be composed of an arbitrary number of steps and different techniques. This is represented in a single, string-based column inside the database. This column can be very simple or very complex, depending on the specific entry.

[7]For example, "NAN" or "NaN" can be mistaken for the acronym for a possible structure group

automatic way, it is also unclear how only a specific part of a procedure can be missing (is it an industrial secret? Is it not reported in the original article? Or maybe the procedure is actually fully reported but one of the authors of the database missed to report this information?)

Differently from other kinds of data, finding a statistical way to impute missing data with a reasonable accuracy is very hard in this case. Accordingly, one of the motivations of our work is the identifications of features that can impact the final performance of a solar cell, and trying to fill the missing data using already known information about other device with similar (yet different) characteristics and performance can hinder the effectiveness of this process. Moreover, many of the features within this database are directly related to specific fabrication and environmental conditions, which are obviously not reported within the database (nor in the original articles) since they are nearly impossible to accurately be measured or, even worse, to be accurately monitored during an extended period of time[8]; for these reasons, classical imputation techniques are not well suited for our case, making it harder to actually enforce all the knowledge provided by the database.
In the next sections, we are going to highlight several approaches we used to try to overcome these limitations and problems, starting from the simplest techniques and then introducing more complex approaches.

5.3 An automatic approach to pattern recognition

First thing first, we want to stress again that, in this work, we do not intend to build a model to predict the final performance of a solar cell. The main reason for this were exposed in 5.2, and here we can add more clarification, specific for the selected use case:

[8]For example, the performance of a device is suspected to be strongly dependent on the humidity, temperature and general weather conditions of the external environment, both during fabrication and usage.

1. From a physical perspective, the performance of a real-world solar cell is influenced by a huge number of external parameters which are hard to record and, for sure, are not present in the used database. For this reason, we think that trying to use the known information to predict the performance of new solar cells can be theoretically possible, but would inevitably lead to numerically wrong results.

2. Available data is present for solar cells based on a small fraction of the possible materials, architecture and fabrication procedures. A predictive model based on this kind of data would be very imprecise (if not completely unreliable) when asked to predict the properties of new devices built with completely new materials or architectures. Moreover, predicting good results for known situations, could be useful only to optimize already existing solutions instead of helping to find new ones, that is actually what the field really needs.

For these reasons, we decided to solve a different problem: which of the actually known and reliably recorded features are more related to the final performance of the device? And, more specifically, do the known devices gather in defined subgroups with uniform characteristics and specific performance ranges? Or, are the structural characteristics of devices less related to the overall performance than we might think?
The questions we pose are subtle, but crucial; for example: the composition of the perovskite used and the architecture of the device are considered to be the most crucial features influencing the final performance of the device. If this is true, we might expect that a clustering algorithm, trained using only these two features, would end up creating groups of devices that are uniform with regard to the features themselves. But would the mean performance of these clusters actually reflect the expected behavior? Or maybe there are other conditions that are more important than these two that we may be overlooking? Or, additionally, could it be that the interplay between individual features is as important as the individual features, or maybe even more important?

By getting our dataset automatically partitioned in different clusters and then analyzing the characteristics of these clusters we hope to overcome the possible biases that the research community might have. Two main scenarios can occur: we can provide an additional proof of known results coming from empirical experience or we might find overlooked patterns, helping to unveil some unknown phenomena.

5.3.1 Clustering algorithms

We used two different clustering algorithms: **KMEANS**[92, 105] and **DBSCAN**[35]. We chose to test both algorithms because they have a crucial difference: **KMEANS** requires the user to give the number of clusters as an input and learns a distance threshold to use to identify the data entries belonging to each cluster, while **DBSCAN** requires the δ (i.e., the distance threshold) to be given as input and learns the number of clusters. Since what we actually want is avoid injecting any kind of previous bias[9], we tried to find the best approach in an empirical way. We first came out with the idea of using **DBSCAN**, since we do not know the number of clusters while we think that there is no actual minimum distance threshold that identifies different clusters; however, we also decided to use **KMEANS** to have another comparison and also to better study the effect of a different number of clusters in a more controlled fashion than what **DBSCAN** allows to do.

Cluster performance is then defined as the mean performance of the devices belonging to the cluster itself. The performance of the devices is measured using specific values that are known to be good measurement of a solar cell performance; in particular, we chose three of these features, which are introduced in the next section.

[9]We do not have any a-priori knowledge of either of the two values, which is actually one of the motivations of this work

5.3.2 Training workflow

Since our aim is to use ML techniques to understand the physics of the system at hand, we used our dataset to train the model and, at the same time, we applied the learning ability of algorithms to identify clusters of devices with homogeneous performances.
Prior than the actual training phase, we performed some data preprocessing. First of all, we removed some features that had absolutely no entries or very few entries (under 50% of the actual dataset dimension). Once obtained a set of usable features, we then proceeded to perform the training of the selected models on different features subsets, in a complexity ramping fashion. Moreover, we identified a set of main features for each target: we used the Pearson correlation coefficient[153] to sort the features based on their correlation with one of the targets, and then selected the best five ones. We chose to restrict the number of features to make it easier to analyze the results, but nothing prevents us in principle from using all the features that proves to be effective in the training of eventual production models. We ended up with three different sets of five features, one for each target. We tried two major approaches:

1. Using all the selected features at ones, building clusters of the resulting dataset.

2. Clustering using only two features at a time and taking the two best performing clusters, selecting a new pair of feature and clustering again. If the new clustering performs better than the previous one, the next step will use those rsults as the starting point; otherwise, will re-use the previous, best performing clusters. The loop is done on all the possible set of two features drawn from the five previously selected features.

Other than these two "competing" approaches, we also performed a round of training using all the possible pairs of features, furtherly trying to identify the most impacting ones. We tested these approaches for three different targets:

1. Power Conversion Efficiency (**PCE**): the key characteristic of a solar cell is its ability to convert light into electricity. This is known as the Power Conversion Efficiency and is the ratio of incident light power to output electrical power.

2. Open Circuit Voltage ($\mathbf{V}_{oc}$): it is the voltage difference measured between two terminals when no current is drawn or supplied. It is the maximum voltage that is available for drawing out from a solar cell

3. Fill Factor (**FF**): the short-circuit current ($\mathbf{I}_{sc}$) and the $\mathbf{V}_{oc}$ are the maximum current and voltage respectively from a solar cell, respectively. However, at both of these operating points, the power from the solar cell is zero. The **FF** is defined as the ratio of the maximum power from the solar cell to the product of $\mathbf{V}_{oc}$ and $\mathbf{I}_{sc}$. It implies that, in conjunction with $\mathbf{V}_{oc}$ and $\mathbf{I}_{sc}$, contributes to determine the **PCE** and so the true maximum power obtainable from a solar cell[137].

5.4 Data preprocessing and encoding strategies

At this point, we performed another preprocessing step, applied to the categorical and string-based columns; in particular, we tried different encoding procedures. The results obtained using each encoding methodology are presented in section 5.5, while the methologies are explained in the following sections.

5.4.1 Categorical encoding

Firstly, we decided to proceed with the simplest technique to encode the categorical features present in our set of five most-target-related features: categorical encoding. This means that we associate a different category (i.e., an integer number) with each unique entry in those categories. For example:

the features `Perovskite_composition_long_form` and `Cell_architecture` are two of these kinds of columns. Two examples of entries of the former are *CsSnBr2.7I0.3* and *Cs0.05FA0.788GU0.032MA0.129PbBr0.51I2.49*, while two of the latter are *nip* and *pin*. Hypothetically, *CsSnBr2.7I0.3* and *nip* could correspond to category number 1 of columns `Perovskite_composition_long_form` and `Cell_architecture` respectively, while *Cs0.05FA0.788GU0.032MA0.129PbBr0.51I2.49* and *pin* could be mapped to category 2. Clearly, if one column has an entry that appears more than once, like the value "nip" in the "Cell_architecture" column, each occurrence will correspond to the same category. Figure 5.3 shows an example result of this process.

5.4.2 One-hot encoding

The next encoding we tried is another classical, namely one-hot encoding. In this case, instead of assigning a single number to each entry, we insert new columns (one for each entry in the original column); each column will be set to 1 if it is the column corresponding to the specific entry in the original column, otherwise it will be set to 0.
For example, if the original column had only two unique entries like *CsSnBr2.7I0.3* and *Cs0.05FA0.788GU0.032MA0.129PbBr0.51I2.49*, each occurrence of the first one would be mapped to $[1, 0]$ (i.e. the value of the first of the two new columns would be set to 1, and the value of the second one to 0) and each occurrence of the second one would be mapped to $[0, 1]$ (i.e. the value of the first new column would be set to 0, and the value of the second to 1).
This method has the advantage of resulting in vectors which metrics are invariant with regard to the order of the entries in the original dataset, which is a property known to be favorable to clustering algorithms like KMeans. However, due to the specific nature of our dataset, it also result in a very sparse matrix, with many new columns where only one of them has an actual value for each row of the dataset, which instead is known to be detrimental to the learning process[185].

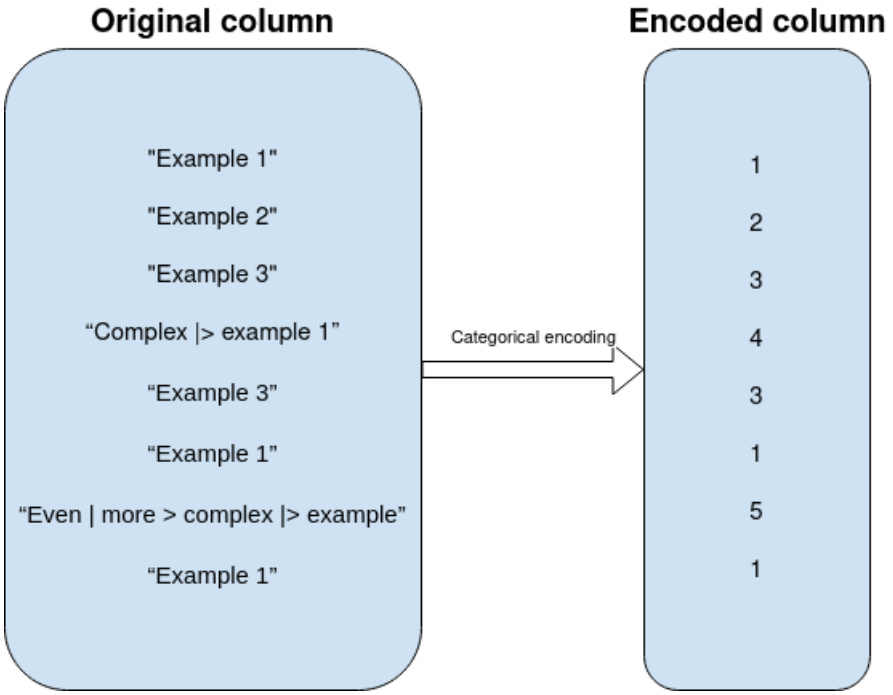


Figure 5.3: A figure showing an example of the application of categorical encoding. We can see how each entry in the original column corresponds to a specific, unique identifier number in the encoded column. *Example 1*, in particular, appears three times, and the corresponding number (i.e., 1) appears three times, in the corresponding positions. The same is true for *Example 3*, which appears two times.

5.4.3 Word tokenization

The second encoding technique we decided to use is word tokenization. In this case, the more complex string-based features have been analyzed in order to identify the characters that convey a sense of order or cause-effect relationship, distinguishing those that were apparently used for spreading different messages and relations. Then, we identified the other set of characters that should be considered as unique entity (for example: the functional

groups present in the perovskite formula[10]) and, consequently, individual words. Once finished, we then used tokenization, and we associated a unique identifier (namely, an integer) to each of the previously identified words. Then, each entry of the target column is transformed into a vector of integers, where each word is transformed into the corresponding integer. Lastly, since not all the resulting vectors have the same length, we took the length of the longest one and padded all vectors to this new length, adding extra zeros[11] where needed.
Figures 5.3 and 5.4 show each step of this process.

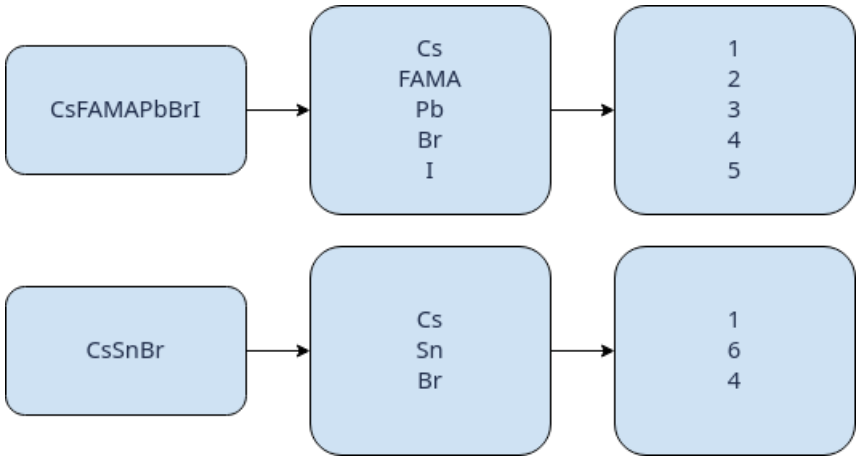


Figure 5.4: Representation of how tokens are assigned. The first example shows how the entry *CsFAMAPbBrI* is splitted in different words, and how each word is assigned a different token; the second example shows how the entry *CsSnBr* is splittend and then the tokens assigned. It must be noted how the words *Cs* and *Br* are assigned to the same two tokens, namely 1 and 4 respectively, in both examples.

[10]For example the entry *CsSnBr* in the column `Perovskite_composition_short_form` should be divided into three different words (one for each element that build the perovskite), but the entry *CsFAMAPbBrI* has 5 words: *Cs* (the caesium element), *Pb* (lead), *Br* (bromine), *I* (iodine), and *FAMA*, which is a more complex group often used for building perovskites.

[11]Obviously, 0 is not used in the previous encoding as part of the translation dictionary.

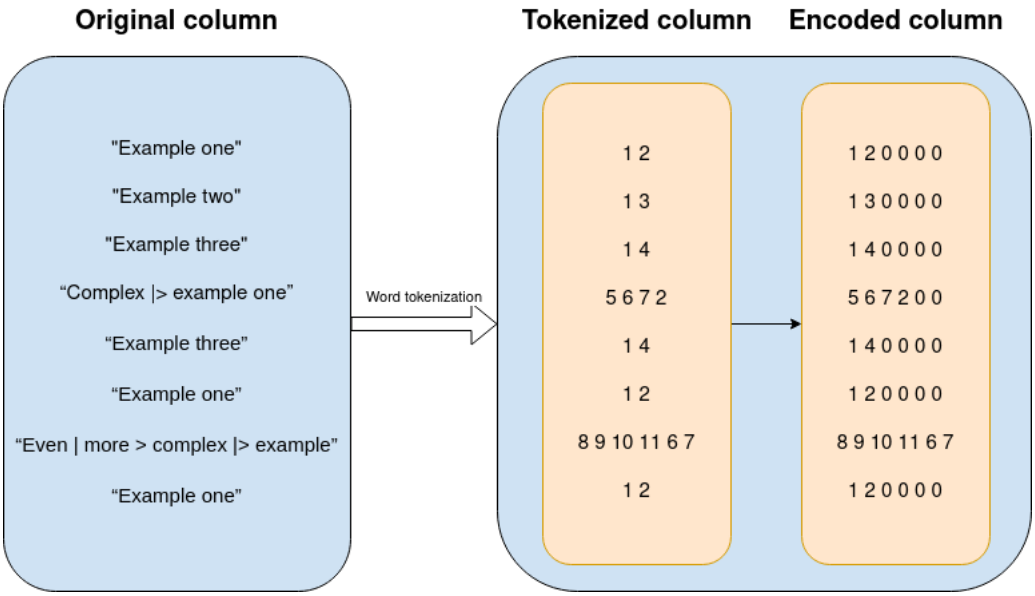


Figure 5.5: Here, we show how the whole tokenization process would work in a synthetic scenario. The leftmost block shows the dataset, and the first orange block shows how each entry is associated with a list of tokens. The second orange block shows the same tokens lists after padding, used to make all of them of the same length (in this case, 6, as the longest tokens vector [8, 9, 10, 11, 6, 7]

While being better than one-hot encoding in this regard, word tokenization still results in sparse matrix.

5.4.4 Multy-hot encoding

This is a sort of combination between Word tokenization and one-hot encoding: we convert each entry of the original column into its corresponding tokens, and then we insert a new column for each unique token present in the whole tokenized column; then, each column corresponding to a token present in the original entry will be set to 1, while the other to 0.

5.4.5 Dimensionality reduction

Since the most likely issue of tokenization is the high dimensionality of the resulting tokenized features, which also ends up being a sparse matrix, we tried to see if performing dimensionality reduction on this big, sparse matrix can lead to better clustering results.
In particular, we chose to use a new, yet very powerful algorithm for dimensionality reduction: **UMAP** (**U**niform **M**anifold **A**pproximation and **P**rojection)[109]; **UMAP** is based on the idea of preserving the global structure of the data while also reducing the dimensionality by projecting the data into a lower-dimensional space being careful about preserving the topology of the original dataset. UMAP uses a combination of techniques:

- **Manifold learning**: this is a technique used to identify patterns in high-dimensional data that can be represented in a lower-dimensional space. In particular, **UMAP** uses a technique called *Riemannian geometry*[172] to model the data as a low-dimensional manifold embedded in a high-dimensional space.

- **Topological data analysis**: is a field of mathematics that studies the global structure of data by analysing its connected components, holes, and other topological features. **UMAP** uses a technique called *simplicial complex*[161] to model the topology of the data. More precisely, **UMAP** uses *fuzzy topological simplicial complex*[161] to compute the probability of two points being connected, and this allows to capture more accurate information about the underlying structure of the data.

- Moreover, **UMAP** uses an optimization algorithm to find the low-dimensional representation that best preserves the distances between the points in the high-dimensional space. This is done by minimizing a loss function that measures the difference between the distances in the high-dimensional space and the distances in the low-dimensional space.

It must be noted that, using the aforementioned techniques, **UMAP** is a non-linear dimensionality reduction algorithm and can handle non-linear struc-

ture of the data. For these reasons, **UMAP** is particularly well suited for being used to create visualization of high dimensional data, allowing for the creation of highly detailed and accurate visualizations of the data, which can be useful for exploring and understanding complex data sets.

5.5 Results

For each of the aforementioned encoding and preprocessing strategies, we are going to show the results of the best clusters obtained and compare them to the baseline of the dataset. We are also going to analyze the values of the features of the devices contained in those clusters, in order to see if these values are homogeneous or not (cfr 5.2).

5.5.1 Results using Categorical encoding and One-hot encoding

First of all, we analyze the two most basic encoding strategies.
In table 5.1 we can see the best overall clustering results obtained using categorical enconding for each target[12]. In table 5.2 we can see the same results obtained using one-hot encoding.

As we can see, for each target both approaches resulted in at least one cluster with significantly higher mean performance than the average of the entire dataset. In particular, the best **PCE**-oriented clusters shows a mean that is nearly double of that of the entire dataset. However, the one-hot encoding approach is generally less performant than categorical encoding.
While it must be noted that these results are all coming from using all the 5 selected features for each target, also the other two approaches resulted in an increase in mean performance of the devices in the best clusters.

[12]More results can be found in Appendix B.

Target	Mean	Baseline	Gain
PCE	22.87	12.0307	190.097
V_{oc}	1.31937	0.961676	137.195
FF	0.826167	0.649566	127.187

Table 5.1: Results of the best performing clusters obtained with categorical encoding for each target considered. The V_{oc} is measured in Volts (V), while PCE is a percentage and FF is a ration between 0 and 1.

Target	Mean	Baseline	Gain
PCE	21.12	12.0307	175.55
V_{oc}	1.2145	0.961676	126.293
FF	0.8141	0.649566	125.323

Table 5.2: Results of the best performing clusters obtained with one-hot encoding for each target considered. The V_{oc} is measured in Volts (V), while PCE is a percentage and FF is a ration between 0 and 1.

It must be noted that these three results have been obtained by the **DBSCAN** algorithm. In general, looking at the bare mean performance and/or the gain in performance of the best clusters with regard to the baseline, the **DBSCAN** algorithm shows higher performance all around for all the three targets using both encoding approaches. However, also the best **KMEANS** best clusters perform better than the baseline, while showing another useful trait, namely a higher number of devices falling within the same, better-performing clusters. For example, the three best performing clusters for the **PCE** target obtained using the **DBSCAN** algorithm and categorical encodings contains, in total, 19 devices. The top three clusters for the same target obtained through **KMEANS** and categorical encodings contains 1017 devices. In particular, the top clusters of the two algorithms contain 8 and 20 devices respectively. For one-hot encoding, the situation is analogous.

While higher performance is the main objective of this work, it should also be noted that too small clusters may end up being statistical anomalies or general outliers of the dataset; on the contrary, bigger clusters allow for more statistically sound conclusions.
However, this higher number of elements in a cluster also implies another crucial difference: a higher number of different feature values in the cluster. For example, the best cluster obtained using **DBSCAN** has all devices with exactly the same feature value for all the 5 used features, while in the best cluster obtained using the **KMEANS** algorithm, one of the features (namely `Perovskite_composition_long_form_uniques`) has 320 different values. It is hard to determine if this is a good or bad quality. For example, it could be better to have more uniform clusters that clearly state that specific values of specific features lead to better performance; on the other hand, this could be the result of human biases that ended up influencing the algorithm, or could be that the algorithm is failing in understanding deeper links between the features and the device performance. While a definitive answer could be found only after a targeted experimental or simulation campaign, we can try to find some indications in the future sections, where we analyze other techniques used trying to obtain better results.

5.5.2 Results using Word tokenization and Many-hot encoding

In table 5.3 we can see the best global results obtained using word tokenization for each target[13]. In table 5.4 we can see the results obtained using may-hot encoding.

In this case, the two approaches obtain literally the same performances for the best clusters. In fact, these clusters actually contains the same devices. Less performant clusters are, instead, slightly different.
Morever, we can notice that the top results are inferior than those obtained

[13]Again, more results can be found in Appendix B.

Target	Mean	Baseline	Gain
PCE	21.494	12.0238	178.76
V_{oc}	0.8262	0.6496	127.185
FF	1.3194	0.9621	137.1394

Table 5.3: Results of the best performing clusters obtained with word tokenization, divided by target. Again, the $\mathbf{V}_{oc}$ is measured in **Volts** (**V**), while **PCE** and **FF** are percentages.

Target	Mean	Baseline	Gain
PCE	21.494	12.0238	178.76
V_{oc}	0.8262	0.6496	127.185
FF	1.3194	0.9621	137.1394

Table 5.4: Results of the best performing clusters obtained with many-hot encoding, divided by target. As always, the $\mathbf{V}_{oc}$ is measured in **Volts** (**V**), while **PCE** and **FF** are percentages.

in the previous experiment; also, the worst performing clusters have higher mean performance than those obtained in the previous experiment, furtherly highlighting how tokenization leads to worse "classification" of the devices. These are again obtained using **DBSCAN**, and **KMEANS** usually have lower performance. However, in this second experiment, using **KMEANS** we obtained clusters containing devices with more uniform features. For example, the best cluster obtained for the **PCE** using categorical encoding and **KMEANS** as the clustering algorithm contained devices with 320 different values in the `Perovskite_composition_long_form_entries` column, while the same cluster obtained using tokenization contains only 269 different values for the same column. This effect is even stronger in the second best performing cluster, where we passed from 326 different values of categorical encoding to 232 of word tokenization.
The lower performance is probably due to the fact that the encoding resulting from word tokenization and many-hot encoding is a sparse matrix, just like one-hot encoding. This sparsity counterbalances (and actually overcomes) the higher expressivity of these two last approaches. However, this higher expressivity has some effect on the **KMEANS** approach, which results in less homogeneous clusters.

5.5.3 Results using UMAP

Differently from simple tokenization, the application of **UMAP** has mixed results. In some cases it lends to better results (in particular for $\mathbf{V}_{oc}$ and **PCE**, both when using **DBSCAN** and **KMEANS**) than tokenization and categorical encoding, while in other it ends up performing worse than tokenization (when applied to **FF**, and again this is true both for **DBSCAN** and **KMEANS**). Another surprising thing is that rising the dimension of the space encoded through **UMAP** does not led always to better performance; for example, when clustering with respect to $\mathbf{V}_{oc}$ using **DBSCAN**, the best performing projection is a 3D space, while all the test done with higher dimensional spaces (namely 5, 8 and 22-D spaces) resulted in pro-

gressively worse results as the dimensionality of the space grows. Finally, in some cases the dimensionality of the space yields no difference in results at all, like when clustering for **PCE** or **FF** using **DBSCAN**. In this case, we obtain nearly the same mean performance for all the different dimensions tried, which could be an indication of the fact that we reached the limit of this approach. However, it must be noted that this approach works better than tokenization for **PCE**, but worse for **FF**.
In table 5.5 we report, as usual, the best performing clusters.

Target	Mean	Baseline	Gain
PCE	25.220	12.0238	209.7509
V_{oc}	0.806	0.6496	124.0802
FF	1.365	0.9621	141.8817

Table 5.5: Results of the best performing clusters obtained with **UMAP** applied to tokenized columns, divided by target. It should be noted that the gain of the **FF** target using tokenization was 127.185, and that the highest score for the **FF** target has been obtained using a 3-dimensional space. As in the previous tables, the $\mathbf{V}_{oc}$ is measured in **Volts** (**V**), **PCE** and **FF** are percentages.

5.6 Discussion

This chapter has been devoted to presenting the activities related to the application of data-science methods to experimental data. We focused on various limitations that usually arise when dealing with hand-collected data coming from many different experiments. As said at the end of chapter 2 and in chapter 3, the lack of standards for data sharing severely impairs the ability to use machine learning pipelines due to the poor quantity and quality of data usually available. In this chapter, we tested some techniques in order to overcome this limitations. In particular, we considered the Per-

ovskite Database, a recently published dataset containing all the available data about perovskite solar cells, i.e. a family of solar cells based on a specific class of materials called perovskite. This dataset has been created by manually collecting all the data contained in all the available papers written on this specific topic, which has been then reunited in a single location. This process is prone to human error, and the data engineering of the more complex information and features is limited. Moreover, not all papers report the same quantities, measurement and characteristics of the discussed devices. The measurement techniques and procedures also vary, making it harder to build a uniform dataset.

Here, we analysed techniques that can potentially deal with these limitations, starting from the selection of sensible performance measurement to use as targets of our evaluation, the selection of a limited number of meaningful features and then furtherly select different feature sets for each target, choosing features which are more related to each specific performance measure.

Finally, we proposed the encoding of complex textual features describing multi-step processes that researchers usually carry out to fabricate devices or materials. These features usually contain many pieces of information, related one to another by a causal or temporal relationship, and being able to efficiently represent these characteristics is key to developing a valuable learning algorithm. To this end, we tested different encoding techniques, moving from the simpler ones (i.e. categorical encoding and one-hot encoding), to slightly more complex ones (like word tokenization and many-hot encoding of the tokenized dataset) and then even projecting the previous high dimensional encoding and datasets to lower dimensional spaces using the **UMAP** algorithm. While results about the best encoding strategy are not conclusive, we highlighted patterns and recognised the limitations of each approach, which can potentially be extended to similar workflows or different problems.

Our aim is to use clustering algorithms to find unknown patterns inside the

available data. This could be a fundamental enabler for future investigations of new devices and to understand the still unknown mechanism that can lead very similar devices to have radically different performance and stability. While this pattern recognition process is usually made "manually" by researchers, using experimental scaffolds and contemporary testing of many different devices, we hope to help to speed up this process by guiding the choice of new device to test based on the knowledge acquired through clustering. Identifying more promising clusters with similar characteristics and higher average performance is a first step towards the identification of new device architectures able to outperform the actual state-of-the-art, while also leading to a deeper understanding of the physics that controls the performance of apparently very similar devices. The intrinsically chaotic nature of the problem requires solid statistical and predictive techniques, and clustering and dimensionality reduction are well known techniques that can be used to find underlying structures and patterns in otherwise messy fields and situations.

As a future work, we aim at using this database as another test-bed for assessing the expressiveness of **MAMBO**. We aim at rebuilding the dataset using the semantic assets given by **MAMBO** (and, possibly, those offered by the new ontology for devices discussed in section 4.7). If we manage to do this, we can re-design the structure of the database and make it more easily accessible and used, while also leading the way to a better representation of experimental data.

Conclusions

In this thesis, we showed the effectiveness of data-driven technologies applied to materials modelling and devices development.
First of all, we outlined the actual cutting edge technologies and methods used both in laboratories to fabricate real-world advanced materials and devices and to fully enforce the power of contemporary HPC clusters to perform multiscale simulations and gather knowledge about materials properties and devices performance. We introduced the physics and chemistry of organic semiconductors, showing the differences that characterize them when compared to traditional silicon-based materials, their advantages and disadvantages, the most active line of research and the techniques used to synthesize them and to use them to fabricate full scale devices like transistors and solar cells. Also, we introduced basic knowledge about the functioning of those classes of devices.
By introducing these state-of-the-art techniques, we also highlighted their limitations and drawbacks, pointing to where there is more room for improvement and tuning or where current technologies simply fail in delivering new results and deeper understanding of scientific problems. Then, we discussed about the benefits that the introduction of data-driven technologies can have for the discipline, the impact that this new approach can have on the whole field and how the integration of experimental workflows, computational simulations and data science and machine learning can lead to improved research activities and new scientific breakthroughs.

We then introduced our first work, devoted to developing a system able to predict properties of a material at the quantistic level. We introduced the problem related to the specific property analyzed (namely, the electronic coupling in organic semiconductor materials) and we discussed the standard approach of computational simulations. Then, we showed how enforcing pre-existing physical knowledge about the analyzed phenomena to chisel machine learning models data features can lead to incredibly well performing models while also achieving a computing time that is orders of magnitude lower than that obtainable using computer simulations or even big deep learning models. Moreover, we showed how this approach can be easily generalized and applied to different molecules and materials and also to predict similar but different properties without changing any step of the pipeline. Even though, at this stage, training is required, training this kind of models is a relatively cheap computation, and it is still more convenient than conventional simulations since they require the same amount of time for each specific case, while the ML model, once trained, can predict new cases in a few milliseconds.
These kinds of models can serve as fundamental enablers for nearly-real-time multiscale investigation of physical processes, allowing for a tighter integration of phenomenon and properties emerging at different scales in order to obtain more accurate and theoretically sound predictions. In particular, our model is able to determine the value of a property for very small aggregates (namely: a molecular pair) and then we can use very efficient statistical methods (like Monte Carlo methods) to calculate the corresponding property of the actual molecular aggregate; this way, we are actually performing a first scale transition very efficiently, allowing us to compute a sort of mean property of a bulk of materials using the corresponding property of the individual particles that build the materials, which is a huge improvement in precision and theoretical soundness. The main drawback of this approach is that it needs a deep understanding of the physical laws that govern the system at hand, which then have to be carefully encoded in machine-indigestible features. While the introduction of deep learning methods made featurization

a thing of the past, we think that it still has a role in the scientific realm: other than the already discussed performance gain, using well humanly understandable features to fit the problem can help researchers to understand how the machine learning models they use work and are able to fit the problem at hand, which can be a key element in allowing them to discover or better understand new physical laws and processes that were previously overlooked.

While the previous work ended up severely outperforming previous approaches, we still faced a huge limitation, namely the very limited availability of data about the analyzed problems. Moreover, very often the available datasets are also uncurated, incomplete and unstandardized, leaving to the data practitioner the burden of developing complex and time consuming pre-processing pipelines to make the data usable. To tackle this problem, we started the development of a new ontology, specifically crafted to deal with molecular materials and related entities and concepts. While ontologies are a well established technology for formally organizing knowledge related to specific domain, they are also very well suited to be the basic ingredient for the standardization of data formats, and they can be used to develop open standards that developers and researchers can then target or enforce when developing software or collecting and sharing data. Moreover, the formal definition of concepts allows for easier application of PSMs to a target question, which can serve both as a way to share common recipes for solving specific classes of problems and as another ingredient in the automation of pipelines.
This kind of research line can be seen as a general endeavor towards the collection of huge amounts of data on the field of materials science. These data are intrinsically unstructured and their collection and usage is also intrinsically dependent on the specific activities that they are used for. This problem is impossible to be solved using classical relational database technologies, and in fact we are already witnessing the application of **NoSQL** technologies to scientific research data. However, even standard **NoSQL** technologies are not suitable to be applied to this specific realm, and researchers are devel-

oping new technologies that are more akin to the concept of data-lakes. In this picture, semantic technologies (like ontologies) that formally associate data and metadata without forcing artificial constructs but only enforcing the intrinsic nature of the data at hand are a fundamental asset.

The next part of our work focused on enforcing techniques similar to those used for molecular properties to whole devices. This activity, which is a joint effort with researchers at the university of Durham, aimed at using computational tools to create a dataset of devices that can then be used to train a ML model to predict properties of the device. In particular, we strove to use only the JV curve as the feature since this is the easiest property to measure also on real-world devices. This way, we can develop ML applications that can also be used for validating experimental data and results, furtherly mixing computational science, experimental science and data science in the research process. Moreover, we also made the whole process that goes from the definition of the device architecture to the final, trained model as automated as possible, aiming at making data science and artificial intelligence more easily adopted from a wider audience of researchers.
We managed to develop a solid approach, training a set of models to predict the carrier mobility of the device (both the overall mean and a directional component) in many different situations, with increasing difficulty. We started from a simple device, ranging only the basic electronic materials parameters (like the mobility itself) and testing both electrons devices and holes devices. Then we did the same while also changing the temperature of the environment, still managing to fit the mobility and even without giving the temperature as an explicit feature to the model. Further tests were performed by varying the thickness of the semiconductor and the channel width of the device, and our model still managed to perfectly fit the mobility and, again, without using these variable parameters as features. Finally, we used a dataset where more parameters were changing at the same time, still managing to get a very high performing model. We then tried to predict other

parameters that were more related to the specific materials used and/or to the geometrical features of those materials. In that case, we did not managed to develop a model able to predict those properties; other than being an expected behavior, this is also a perfect example of where the approach outlined in the second chapter (and more generally, in this thesis) can be a game changer: notably, we can use a model akin to the one obtained in chapter 2 to compute the properties of the materials used to build the device, and then use this predicted properties to help training another model which is then able to correctly predict the characteristics of the full device. While this approach is still theoretically possible using computational simulations, the computational resources needed for simulations and the long effort required to organize an actual simulation campaign makes it materially unfeasible. However, with the necessary preparation, we can easily develop a model specifically trained to predict the properties we need and then use this model to move from atomistic or molecular properties to those of devices built with the chosen materials.

In the last chapter, we also showed how to work on data directly coming from real-world devices, and in particular perovskite solar cells. In particular, we wanted to use automatic approaches to find unknown patterns in already available data in order to give researchers indications on the type of devices and materials to test in order to obtain solar cells with higher performances. Again, we had to develop techniques to overcome the limitations of the available dataset. In particular, the whole collection process has been done manually by the authors of the dataset, which is prone to human error. Moreover, the lack of uniformity in data presentation of the original papers led to very inconsistent formats[14] and incomplete entries. Moreover, the high amount of missing data forced us to drop many features since there is no clear

[14]For example, the authors used many different strings for missing data; the dataset contains many features with the same type of information but with different encodings; the dataset contains very complex features without any kind of explanation about the meaning of specific characters used to communicate an idea of causal or temporal relationships.

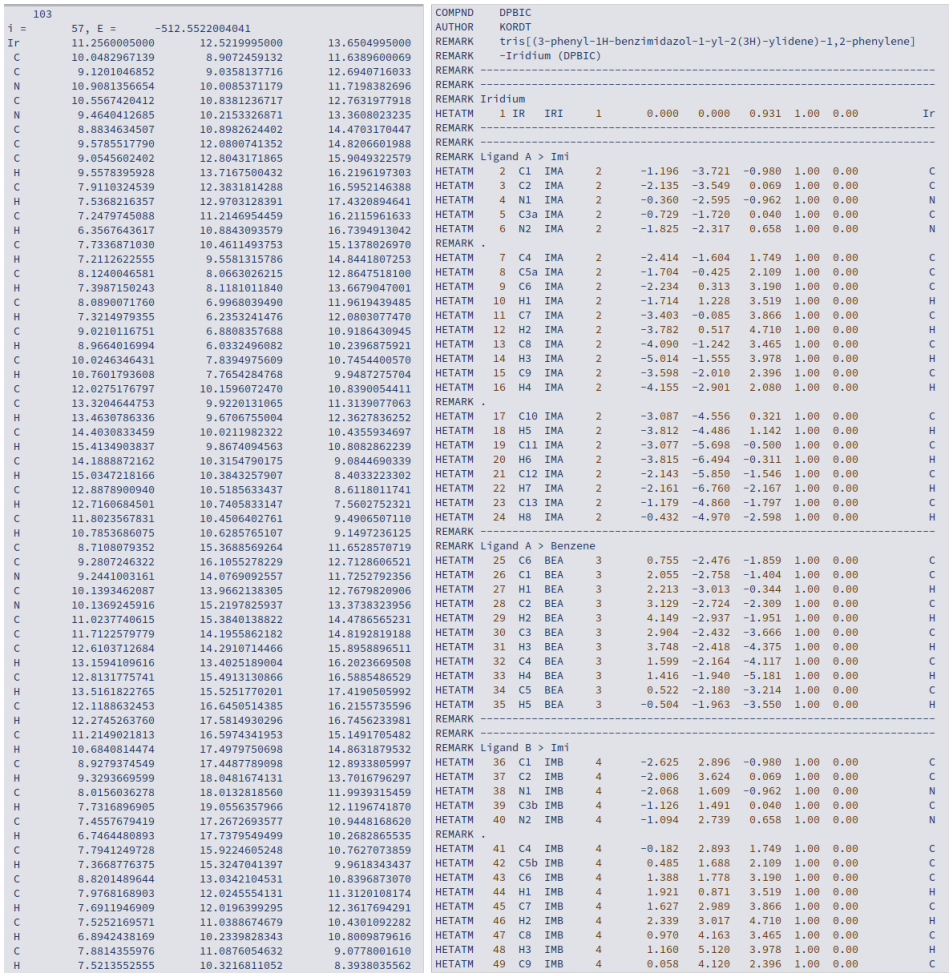
method on how to statistically or automatically infer sensible values.
For these reasons, we decided to perform clusterization on three different measurements that are used to judge the quality of a solar cell: the **PCE**, the **FF** and the $\mathbf{V}_{oc}$. First of all, we identified 5 different features that are statistically more related to each of these targets using the Pearson correlation, selecting them between those with a reasonable amount of non-missing data. Then, we tried different approaches for the encoding of the string-based, complex features in order to make them more easily processed by the learning algorithm. In particular, we tried categorical encoding, one-hot encoding, word tokenization and many-hot encoding of the tokens, and we saw how the simpler approach outperformed the supposedly more expressive ones. Since we thought that this is caused by the sparse nature of the datasets resulting from the more complex encodings, we used another algorithm, namely **UMAP**, to project the tokenized features to a lower-dimensional space, resulting in a non-sparse matrix. This process actually achieved a better performance than categorical encoding on two targets.

All these methods and research lines show how data-driven technologies can help to make it easier and faster to gain insights using historical data. We gave different examples on how already existing datasets and easily usable methodologies coming from data-science can lead to new insights and improved predictions, while allowing for tighter integration between experimental data, computational science and bridging the gap between knowledge coming from different scales. We also showed how semantic technologies can furtherly serve the discipline by creating common platforms for data sharing and standardization, which can immensely improve the reliability and applicability of the aforementioned data-driven procedures. While prototypical, these techniques showed promising results and leave huge room for improvement in future work.

Appendix A

Chemical files conversion based on MAMBO

Here we leave some figures that, for ease of readability, could not fit in chapter 3.



```
    103
i =       57, E =      -512.5522004041
Ir          11.2560005000       12.5219995000       13.6504995000
 C          10.0482967139        8.9072459132       11.6389600069
 C           9.1201046852        9.0358137716       12.6940716033
 N          10.9081356654       10.0085371179       11.7198382696
 C          10.5567420412       10.8381236717       12.7631977918
 N           9.4640412685       10.2153326871       13.3608023235
 C           8.8834634507       10.8982624402       14.4703170447
 C           9.5785517790       12.0800741352       14.8206601988
 C           9.0545602402       12.8043171865       15.9049322579
 H           9.5578395928       13.7167500432       16.2196197303
 C           7.9110324539       12.3831814288       16.5952146388
 H           7.5368216357       12.9703128391       17.4320894641
 C           7.2479745088       11.2146954459       16.2115961633
 H           6.3567643617       10.8843093579       16.7394913042
 C           7.7336871030       10.4611493753       15.1378026970
 H           7.2112622555        9.5581315786       14.8441807253
 C           8.1240046581        8.0663026215       12.8647518100
 H           7.3987150243        8.1181011840       13.6679047001
 C           8.0890071760        6.9968039490       11.9619439485
 H           7.3214979355        6.2353241476       12.0803077470
 C           9.0210116751        6.8808357688       10.9186430945
 H           8.9664016994        6.0332496082       10.2396875921
 C          10.0246346431        7.8394975609       10.7454400570
 H          10.7601793608        7.7654284768        9.9487275704
 C          12.0275176797       10.1596072470       10.8390054411
 C          13.3204644753        9.9220131065       11.3139077063
 H          13.4630786336        9.6706755004       12.3627836252
 C          14.4030833459       10.0211982322       10.4355934697
 H          15.4134903837        9.8674094563       10.8082862239
 C          14.1888872162       10.3154790175        9.0844690339
 H          15.0347218166       10.3843257907        8.4033223302
 C          12.8878900940       10.5185633437        8.6118011741
 H          12.7160684501       10.7405833147        7.5602752321
 C          11.8023567831       10.4506402761        9.4906507110
 H          10.7853686075       10.6285765107        9.1497236125
 C           8.7108079352       15.3688569264       11.6528570719
 C           9.2807246322       16.1055278229       12.7128606521
 N           9.2441003161       14.0769092557       11.7252792356
 C          10.1393462087       13.9662138305       12.7679820906
 N          10.1369245916       15.2197825937       13.3738323956
 C          11.0237740615       15.3840138822       14.4786565231
 C          11.7122579779       14.1955862182       14.8192819188
 C          12.6103712684       14.2910714466       15.8958896511
 H          13.1594109616       13.4025189004       16.2023669508
 C          12.8131775741       15.4913130866       16.5885486529
 H          13.5161822765       15.5251770201       17.4190505992
 C          12.1188632453       16.6450514385       16.2155735596
 H          12.2745263760       17.5814930296       16.7456233981
 C          11.2149021813       16.5974341953       15.1491705482
 H          10.6840814474       17.4979750698       14.8631879532
 C           8.9279374549       17.4487789098       12.8933805997
 H           9.3293669599       18.0481674131       13.7016796297
 C           8.0156036278       18.0132818560       11.9939315459
 H           7.7316896905       19.0556357966       12.1196741870
 C           7.4557679419       17.2672693577       10.9448168620
 H           6.7464480893       17.7379549499       10.2682865535
 C           7.7941249728       15.9224605248       10.7627073859
 H           7.3668776375       15.3247041397        9.9618343437
 C           8.8201489644       13.0342104531       10.8396873070
 C           7.9768168903       12.0245554131       11.3120108174
 H           7.6911946909       12.0196399295       12.3617694291
 C           7.5252169571       11.0388674679       10.4301092282
 H           6.8942438169       10.2339828343       10.8009879616
 C           7.8814355976       11.0876054632        9.0778001610
 H           7.5213552555       10.3216811052        8.3938035562
```

```
COMPND    DPBIC
AUTHOR    KORDT
REMARK    tris[(3-phenyl-1H-benzimidazol-1-yl-2(3H)-ylidene)-1,2-phenylene]
REMARK    -Iridium (DPBIC)
REMARK ------------------------------------------------------------------------
REMARK ------------------------------------------------------------------------
REMARK Iridium
HETATM    1 IR   IRI     1       0.000   0.000   0.931  1.00  0.00           Ir
REMARK ------------------------------------------------------------------------
REMARK ------------------------------------------------------------------------
REMARK Ligand A > Imi
HETATM    2  C1  IMA     2      -1.196  -3.721  -0.980  1.00  0.00           C
HETATM    3  C2  IMA     2      -2.135  -3.549   0.069  1.00  0.00           C
HETATM    4  N1  IMA     2      -0.360  -2.595  -0.962  1.00  0.00           N
HETATM    5  C3a IMA     2      -0.729  -1.720   0.040  1.00  0.00           C
HETATM    6  N2  IMA     2      -1.825  -2.317   0.658  1.00  0.00           N
REMARK .
HETATM    7  C4  IMA     2      -2.414  -1.604   1.749  1.00  0.00           C
HETATM    8  C5a IMA     2      -1.704  -0.425   2.109  1.00  0.00           C
HETATM    9  C6  IMA     2      -2.234   0.313   3.190  1.00  0.00           C
HETATM   10  H1  IMA     2      -1.714   1.228   3.519  1.00  0.00           H
HETATM   11  C7  IMA     2      -3.403  -0.085   3.866  1.00  0.00           C
HETATM   12  H2  IMA     2      -3.782   0.517   4.710  1.00  0.00           H
HETATM   13  C8  IMA     2      -4.090  -1.242   3.465  1.00  0.00           C
HETATM   14  H3  IMA     2      -5.014  -1.555   3.978  1.00  0.00           H
HETATM   15  C9  IMA     2      -3.598  -2.010   2.396  1.00  0.00           C
HETATM   16  H4  IMA     2      -4.155  -2.901   2.080  1.00  0.00           H
REMARK .
HETATM   17  C10 IMA     2      -3.087  -4.556   0.321  1.00  0.00           C
HETATM   18  H5  IMA     2      -3.812  -4.486   1.142  1.00  0.00           H
HETATM   19  C11 IMA     2      -3.077  -5.698  -0.500  1.00  0.00           C
HETATM   20  H6  IMA     2      -3.815  -6.494  -0.311  1.00  0.00           H
HETATM   21  C12 IMA     2      -2.143  -5.850  -1.546  1.00  0.00           C
HETATM   22  H7  IMA     2      -2.161  -6.760  -2.167  1.00  0.00           H
HETATM   23  C13 IMA     2      -1.179  -4.860  -1.797  1.00  0.00           C
HETATM   24  H8  IMA     2      -0.432  -4.970  -2.598  1.00  0.00           H
REMARK ------------------------------------------------------------------------
REMARK Ligand A > Benzene
HETATM   25  C6  BEA     3       0.755  -2.476  -1.859  1.00  0.00           C
HETATM   26  C1  BEA     3       2.055  -2.758  -1.404  1.00  0.00           C
HETATM   27  H1  BEA     3       2.213  -3.013  -0.344  1.00  0.00           H
HETATM   28  C2  BEA     3       3.129  -2.724  -2.309  1.00  0.00           C
HETATM   29  H2  BEA     3       4.149  -2.937  -1.951  1.00  0.00           H
HETATM   30  C3  BEA     3       2.904  -2.432  -3.666  1.00  0.00           C
HETATM   31  H3  BEA     3       3.748  -2.418  -4.375  1.00  0.00           H
HETATM   32  C4  BEA     3       1.599  -2.164  -4.117  1.00  0.00           C
HETATM   33  H4  BEA     3       1.416  -1.940  -5.181  1.00  0.00           H
HETATM   34  C5  BEA     3       0.522  -2.180  -3.214  1.00  0.00           C
HETATM   35  H5  BEA     3      -0.504  -1.963  -3.550  1.00  0.00           H
REMARK ------------------------------------------------------------------------
REMARK ------------------------------------------------------------------------
REMARK Ligand B > Imi
HETATM   36  C1  IMB     4      -2.625   2.896  -0.980  1.00  0.00           C
HETATM   37  C2  IMB     4      -2.006   3.624   0.069  1.00  0.00           C
HETATM   38  N1  IMB     4      -2.068   1.609  -0.962  1.00  0.00           N
HETATM   39  C3b IMB     4      -1.126   1.491   0.040  1.00  0.00           C
HETATM   40  N2  IMB     4      -1.094   2.739   0.658  1.00  0.00           N
REMARK .
HETATM   41  C4  IMB     4      -0.182   2.893   1.749  1.00  0.00           C
HETATM   42  C5b IMB     4       0.485   1.688   2.109  1.00  0.00           C
HETATM   43  C6  IMB     4       1.388   1.778   3.190  1.00  0.00           C
HETATM   44  H1  IMB     4       1.921   0.871   3.519  1.00  0.00           H
HETATM   45  C7  IMB     4       1.627   2.989   3.866  1.00  0.00           C
HETATM   46  H2  IMB     4       2.339   3.017   4.710  1.00  0.00           H
HETATM   47  C8  IMB     4       0.970   4.163   3.465  1.00  0.00           C
HETATM   48  H3  IMB     4       1.160   5.120   3.978  1.00  0.00           H
HETATM   49  C9  IMB     4       0.058   4.120   2.396  1.00  0.00           C
```

Figure A.1: Two fragments of two different files representing the same molecule (aside from the specific coordinates). On the left, the .xyz file, on the right the .pdb file. While representing the same entity, the contain different explicit information, and even the information contained in both files (namely the coordinates and the atom types) are represented differently.

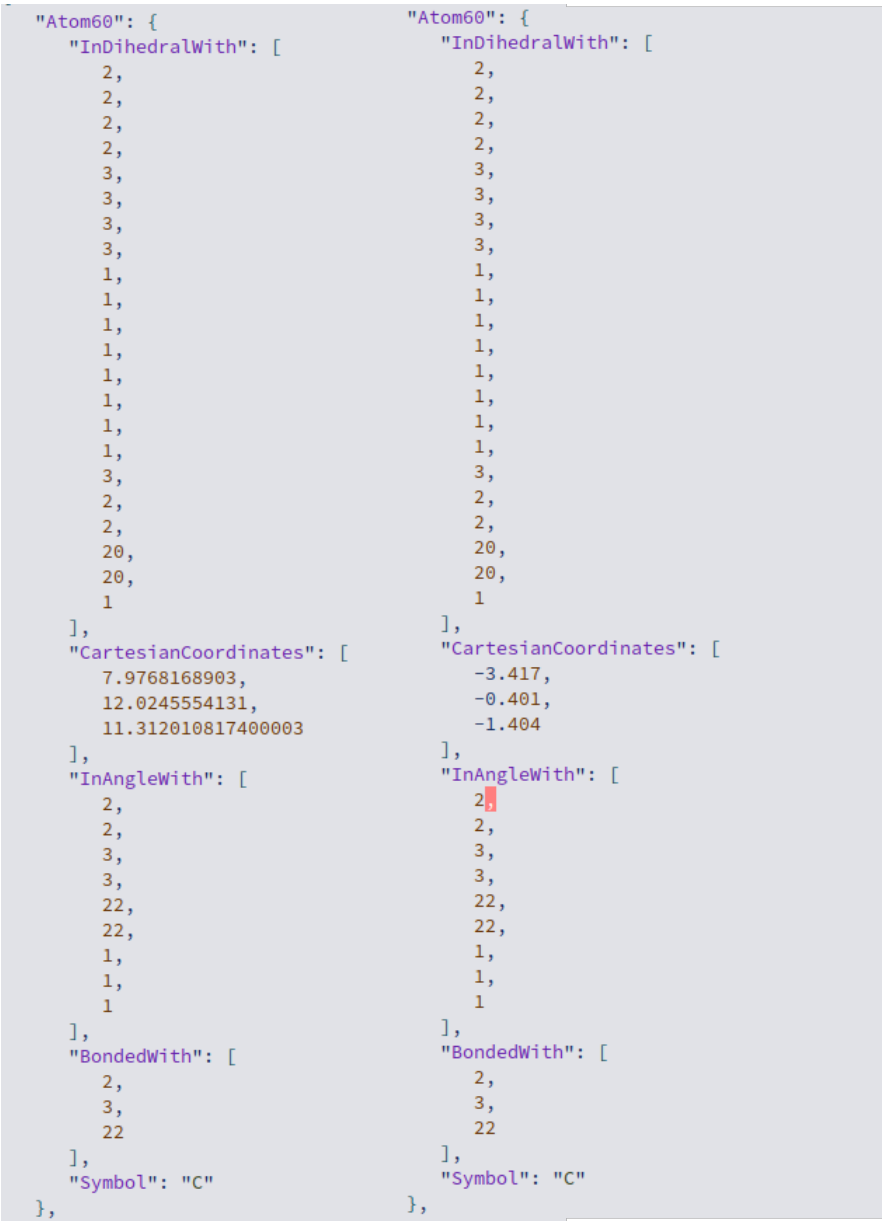


Figure A.2: Excerpts of the two .json files corresponding to those shown in figure A.1; on the left, the one corresponding to the .xyz file, on the right the one corresponding to the .pdb file. We can see how, aside from the specific coordinates, the two files are identical. Moreover, looking at the one built from the .xyz file, it must be noted that we were able to insert more information than those contained in the original file, namely the bonds, the angles and the dihedrals.

Appendix B

Perovskite solar cells clustering results

Here we discuss some more results obtained through clustering of the Perovskite database[1].
First of all, let's look at the results obtained using the round-based approach, meaning using two features at once for the clustering and then using other two features to perform clustering again using the data contained in the two best-performing clusters of the previous round.
Let's see how this approach performed for the **PCE** in table B.2 and B.1.

Round	Mean	Gain
6	14.1187	117.356
7	15.8291	131.573
10	16.7367	139.116

Table B.1: Results for the best rounds using **KMEANS** for the **PCE**. The baseline is 12.0307.

[1]It must be noted that in this appendix on we are going to discuss results obtained only using categorical encoding and word tokenization, since these have been the first methods we tested.

Round	Mean	Gain
1	12.3702	102.822
2	20.8217	173.071
3	20.8217	173.071

Table B.2: Results for the best round using **DBSCAN** for the **PCE**. The baseline is 12.0307. Here we can se that, differently than what happens for **KMEANS** (B.1 the performance reached the top at round 3 and is higher than what we achieved using **KMEANS**.

For the **FF** we have a very different scenario: we obtain better performance using **KMEANS** instead of **DBSCAN**. Even the trend of growth is different: in fact, **DBSCAN** grows slower and grows until the last rounds, while with **KMEANS** we still have lower performances but it reaches the top values during first rounds. It must be noted that this is the only case where **KMEANS** outperforms **DBSCAN** in the entire work.
These results are visible in B.3 and B.4.

Round	Mean	Gain
1	0.699818	107.736
2	0.723625	111.401
3	0.77075	118.656

Table B.3: Results for the best rounds using **KMEANS** for the **FF**. The baseline is 0.649566.

Finally, for the $\mathbf{V}_{oc}$ we have a situation very similar to that seen for the **PCE**, where **KMEANS** performance grows slower than that of **DBSCAN**. Howver, while **DBSCAN** again outperforms **KMEANS**, here the difference in performance between the two approaches is smaller than in the first case. We can see those results in B.5 and B.6.

Round	Mean	Gain
8	0.757333	116.591
9	0.76425	117.655
10	0.76425	117.655

Table B.4: Results for the best round using **DBSCAN** for the **FF**. The baseline is 0.649566.

Round	Mean	Gain
7	0.999612	103.945
9	1.00681	104.693
10	1.06005	110.229

Table B.5: Results for the best rounds using **KMEANS** for the **FF**. The baseline is 0.961676.

Round	Mean	Gain
2	1.08167	112.477
3	1.11217	115.649
4	1.1186	116.318

Table B.6: Results for the best round using **DBSCAN** for the **FF**. The baseline is 0.961676.

Bibliography

[1] 2030, A. M. I. Materials 2030 roadmap. *https://www.ami2030.eu/wp-content/uploads/2022/12/2022-12-09$_{M}aterials_{2}030_{R}oadMap_{V}F4.pdf$.*

[2] Adjogri, S. J., and Meyer, E. L. A review on lead-free hybrid halide perovskites as light absorbers for photovoltaic applications based on their structural, optical, and morphological properties. *Molecules 2020, Vol. 25, Page 5039 25* (10 2020), 5039.

[3] Afgan, E., Nekrutenko, A., Grüning, B. A., Blankenberg, D., Goecks, J., Schatz, M. C., Ostrovsky, A. E., Mahmoud, A., Lonie, A. J., Syme, A., Fouilloux, A., Bretaudeau, A., Nekrutenko, A., Kumar, A., Eschenlauer, A. C., Desanto, A. D., Guerler, A., Serrano-Solano, B., Batut, B., Grüning, B. A., Langhorst, B. W., Carr, B., Raubenolt, B. A., Hyde, C. J., Bromhead, C. J., Barnett, C. B., Royaux, C., Gallardo, C., Blankenberg, D., Fornika, D. J., Baker, D., Bouvier, D., Clements, D., Morais, D. A. D. L., Tabernero, D. L., Lariviere, D., Nasr, E., Afgan, E., Zambelli, F., Heyl, F., Psomopoulos, F., Coppens, F., Price, G. R., Cuccuru, G., Corguillé, G. L., Kuster, G. V., Akbulut, G. G., Rasche, H., Hans-Rudolf, H., Eguinoa, I., Makunin, I., Ranawaka, I. J., Taylor, J. P., Joshi, J., Hillman-Jackson, J., Goecks, J., Chilton, J. M., Kamali, K., Suderman, K., Poterlowicz, K., Yvan, L. B., Lopez-Delisle, L., Sargent, L., Bassetti, M. E., Tangaro, M. A., Beek,

M. V. D., Cech, M., Bernt, M., Fahrner, M., Tekman, M., Föll, M. C., Schatz, M. C., Crusoe, M. R., Roncoroni, M., Kucher, N., Coraor, N., Stoler, N., Rhodes, N., Soranzo, N., Pinter, N., Goonasekera, N. A., Moreno, P. A., Videm, P., Melanie, P., Mandreoli, P., Jagtap, P. D., Gu, Q., Weber, R. J., Lazarus, R., Vorderman, R. H., Hiltemann, S., Golitsynskiy, S., Garg, S., Bray, S. A., Gladman, S. L., Leo, S., Mehta, S. P., Griffin, T. J., Jalili, V., Yves, V., Wen, V., Nagampalli, V. K., Bacon, W. A., Koning, W. D., Maier, W., and Briggs, P. J. The galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2022 update. *Nucleic Acids Research 50* (7 2022), W345–W351.

[4] Andersen, C., Armiento, R., Blokhin, E., Conduit, G., Dwaraknath, S., Evans, M. L., Fekete, A., Gopakumar, A., Gražulis, S., Hegde, V., Horton, M., Merkys, A., Mohamed, F., Morris, A., Oses, C., Pizzi, G., Purcell, T., Rignanese, G.-M., Scheffler, M., Scheidgen, M., Talirz, L., Toher, C., Uhrin, M., Winston, D., and Wolverton, C. The optimade specification.

[5] Andersen, C. W., Armiento, R., Blokhin, E., Conduit, G. J., Dwaraknath, S., Evans, M. L., ádám Fekete, Gopakumar, A., Gražulis, S., Merkys, A., Mohamed, F., Oses, C., Pizzi, G., Rignanese, G. M., Scheidgen, M., Talirz, L., Toher, C., Winston, D., Aversa, R., Choudhary, K., Colinet, P., Curtarolo, S., Stefano, D. D., Draxl, C., Er, S., Esters, M., Fornari, M., Giantomassi, M., Govoni, M., Hautier, G., Hegde, V., Horton, M. K., Huck, P., Huhs, G., Hummelshøj, J., Kariryaa, A., Kozinsky, B., Kumbhar, S., Liu, M., Marzari, N., Morris, A. J., Mostofi, A. A., Persson, K. A., Petretto, G., Purcell, T., Ricci, F., Rose, F., Schef-

fler, M., Speckhard, D., Uhrin, M., Vaitkus, A., Villars, P., Waroquiers, D., Wolverton, C., Wu, M., and Yang, X. Optimade, an api for exchanging materials data. *Scientific Data 2021 8:1 8* (8 2021), 1–10.

[6] Antoniou, G., and van Harmelen, F. Web ontology language: Owl. *Handbook on Ontologies* (2004), 67–92.

[7] Babayigit, A., Ethirajan, A., Muller, M., and Conings, B. Toxicity of organometal halide perovskite solar cells. *Nature Materials 2016 15:3 15* (2 2016), 247–251.

[8] Babayigit, A., Thanh, D. D., Ethirajan, A., Manca, J., Muller, M., Boyen, H. G., and Conings, B. Assessing the toxicity of pb- and sn-based perovskite solar cells in model organism danio rerio. *Scientific Reports 2016 6:1 6* (1 2016), 1–11.

[9] Bagal, V., Aggarwal, R., Vinod, P. K., and Priyakumar, U. D. Molgpt: Molecular generation using a transformer-decoder model. *Journal of Chemical Information and Modeling 62*, 9 (May 2022), 2064–2076.

[10] Baldoni, M., Lorenzoni, A., Pecchia, A., and Mercuri, F. Spatial and orientational dependence of electron transfer parameters in aggregates of iridium-containing host materials for oleds: Coupling constrained density functional theory with molecular dynamics. *Physical Chemistry Chemical Physics 20* (2018), 28393–28399.

[11] Bandrowski, A., Brinkman, R., Brochhausen, M., Brush, M. H., Bug, B., Chibucos, M. C., Clancy, K., Courtot, M., Derom, D., Dumontier, M., Fan, L., Fostel, J., Fragoso, G., Gibson, F., Gonzalez-Beltran, A., Haendel, M. A., He, Y., Heiskanen, M., Hernandez-Boussard, T., Jensen, M., Lin, Y., Lister, A. L., Lord, P., Malone, J., Manduchi,

E., McGee, M., Morrison, N., Overton, J. A., Parkinson, H., Peters, B., Rocca-Serra, P., Ruttenberg, A., Sansone, S. A., Scheuermann, R. H., Schober, D., Smith, B., Soldatova, L. N., Stoeckert, C. J., Taylor, C. F., Torniai, C., Turner, J. A., Vita, R., Whetzel, P. L., and Zheng, J. The ontology for biomedical investigations. *PLOS ONE 11* (4 2016), e0154556.

[12] Baseden, K. A., and Tye, J. W. Introduction to density functional theory: Calculations by hand on the helium atom. *Journal of Chemical Education 91* (12 2014), 2116–2123.

[13] Batchelor, C. Chemical methods ontology.

[14] Benmessaoud, I. R., Mahul-Mellier, A. L., Horváth, E., Maco, B., Spina, M., Lashuel, H. A., and Forró, L. Health hazards of methylammonium lead iodide based perovskites: cytotoxicity studies. *Toxicology Research 5* (3 2016), 407–419.

[15] Berendsen, H. J., van der Spoel, D., and van Drunen, R. Gromacs: A message-passing parallel molecular dynamics implementation. *Computer Physics Communications 91* (9 1995), 43–56.

[16] Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, P. E. The protein data bank. *Nucleic Acids Res 28*, 1 (Jan. 2000), 235–242.

[17] Branco, P., Ribeiro, R. P., Torgo, L., Krawczyk, B., and Moniz, N. Smogn: a pre-processing approach for imbalanced regression. *Proceedings of Machine Learning Research 74* (2017), 36–50.

[18] Bronstein, H., Nielsen, C. B., Schroeder, B. C., and McCulloch, I. The role of chemical design in the performance of or-

ganic semiconductors. *Nature Reviews Chemistry 2020 4:2 4* (1 2020), 66–77.

[19] Bryant, D., Aristidou, N., Pont, S., Sanchez-Molina, I., Chotchunangatchaval, T., Wheeler, S., Durrant, J. R., and Haque, S. A. Light and oxygen induced degradation limits the operational stability of methylammonium lead triiodide perovskite solar cells. *Energy Environmental Science 9* (5 2016), 1655–1660.

[20] Burschka, J., Pellet, N., Moon, S. J., Humphry-Baker, R., Gao, P., Nazeeruddin, M. K., and Grätzel, M. Sequential deposition as a route to high-performance perovskite-sensitized solar cells. *Nature 2013 499:7458 499* (7 2013), 316–319.

[21] Butler, K. T., Davies, D. W., Cartwright, H., Isayev, O., and Walsh, A. Machine learning for molecular and materials science. *Nature 2018 559:7715 559* (7 2018), 547–555.

[22] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research 16* (2002), 321–357.

[23] Chen, C., and Ong, S. P. A universal graph deep learning interatomic potential for the periodic table. *Nature Computational Science 2*, 11 (Nov 2022), 718–728.

[24] Chen, C., Ye, W., Zuo, Y., Zheng, C., and Ong, S. P. Graph networks as a universal machine learning framework for molecules and crystals. *Chemistry of Materials 31*, 9 (May 2019), 3564–3572.

[25] Chen, X., Zhou, H., and Wang, H. 2d/3d halide perovskites for optoelectronic devices. *Frontiers in Chemistry 9* (8 2021), 679.

[26] Commission, E., for Research, D.-G., and Innovation. *Refinding industry : defining innovation.* Publications Office, 2018.

[27] Conwell, E. M. Impurity band conduction in germanium and silicon. *Physical Review 103* (7 1956), 51.

[28] Coropceanu, V., Cornil, J., da Silva Filho, D. A., Olivier, Y., Silbey, R., and Brédas, J. L. Charge transport in organic semiconductors. *Chemical Reviews 107* (4 2007), 926–952.

[29] Cremonesi, A., Grobert, N., Gumbsch, P., Piketty, L., Montelius, L., Vandeputte, K., and Vérilhac, I. Materials 2030 manifesto. *Advanced Materials Initiative 2030 https://www.ami2030.eu/wp-content/uploads/2022/06/advanced-materials-2030-manifesto-Published-on-7-Feb-2022.pdf*.

[30] Degtyarenko, K., matos, P. D., Ennis, M., Hastings, J., Zbinden, M., Mcnaught, A., Alcántara, R., Darsow, M., Guedj, M., and Ashburner, M. Chebi: a database and ontology for chemical entities of biological interest. *Nucleic Acids Research 36* (1 2008), D344–D350.

[31] Diao, Y., Shaw, L., Bao, Z., and Mannsfeld, S. C. Morphology control strategies for solution-processed organic semiconductor thin films. *Energy Environmental Science 7* (6 2014), 2145–2159.

[32] Diemer, P. J., Lyle, C. R., Mei, Y., Sutton, C., Payne, M. M., Anthony, J. E., Coropceanu, V., Brédas, J.-L., Jurchescu, O. D., Diemer, P. J., Lyle, C. R., Mei, Y., Jurchescu, O. D., Sutton, C., Coropceanu, V., Brédas, J. L., Payne, M. M., and Anthony, J. E. Vibration-assisted crystallization improves organic/dielectric interface in organic thin-film transistors. *Advanced Materials 25* (12 2013), 6956–6962.

[33] Draxl, C., and Scheffler, M. Nomad: The fair concept for big data-driven materials science. *MRS Bulletin 43* (9 2018), 676–682.

[34] Elliott, J. A. Novel approaches to multiscale modelling in materials science. *International Materials Reviews 56* (7 2013), 207–225.

[35] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise.

[36] Fang, X., Liu, L., Lei, J., He, D., Zhang, S., Zhou, J., Wang, F., Wu, H., and Wang, H. Geometry-enhanced molecular representation learning for property prediction. *Nature Machine Intelligence 2022 4:2 4* (2 2022), 127–134.

[37] Fleming, I. Molecular orbitals and organic chemical reactions, reference edition. *Molecular Orbitals and Organic Chemical Reactions, Reference Edition* (2 2010).

[38] Forrest, S. R. The path to ubiquitous and low-cost organic electronic appliances on plastic. *Nature 2004 428:6986 428* (4 2004), 911–918.

[39] Fratini, S., Nikolka, M., Salleo, A., Schweicher, G., and Sirringhaus, H. Charge transport in high-mobility conjugated polymers and molecular semiconductors.

[40] Galindo, S., Tamayo, A., Leonardi, F., Mas-Torrent, M., Galindo, S., Tamayo, A., Leonardi, F., and Mas-Torrent, M. Control of polymorphism and morphology in solution sheared organic field-effect transistors. *Advanced Functional Materials 27* (7 2017), 1700526.

[41] Gao, L., Chao, L., Hou, M., Liang, J., Chen, Y., Yu, H. D., and Huang, W. Flexible, transparent nanocellulose paper-based perovskite solar cells. *npj Flexible Electronics 2019 3:1 3* (2 2019), 1–8.

[42] Ghedini, E., and Schmitz, G. Emmo the european materials modelling ontology. *EMMC Workshop on Interoperability in Materials Modelling* (2017), 7–8.

[43] Gomez-Perez, A., and Benjamins, V. R. Applications of ontologies and problem-solving methods. *AI Magazine 20* (3 1999), 119–119.

[44] Gong, J., Darling, S. B., and You, F. Perovskite photovoltaics: life-cycle assessment of energy and environmental impacts. *Energy Environmental Science 8* (7 2015), 1953–1968.

[45] Goto, C. O., Tomiya, S., Murakami, Y., Shinozaki, A., Toda, A., Kasahara, J., Hobara, D., Goto, O., Tomiya, S., Murakami, Y., Hobara, D., Shinozaki, A., Toda, A., and Kasahara, J. Organic single-crystal arrays from solution-phase growth using micropattern with nucleation control region. *Advanced Materials 24* (2 2012), 1117–1122.

[46] Gruber, T. R. A translation approach to portable ontology specifications. *Knowledge Acquisition 5* (6 1993), 199–220.

[47] Habisreutinger, S. N., Leijtens, T., Eperon, G. E., Stranks, S. D., Nicholas, R. J., and Snaith, H. J. Carbon nanotube/polymer composites as a highly stable hole collection layer in perovskite solar cells. *Nano Letters 14* (10 2014), 5561–5568.

[48] Hagelien, T. F., Preisig, H. A., Friis, J., Klein, P., and Konchakova, N. A practical approach to ontology-based data modelling for semantic interoperability. *14th WCCM-ECCOMAS Congress 2020 2100 - Other* (3 2021).

[49] Hakimi, O., Gelpi, J. L., Krallinger, M., Curi, F., Repchevsky, D., and Ginebra, M. P. The devices, experimental scaffolds, and biomaterials ontology (deb): A tool for mapping, annota-

tion, and analysis of biomaterials data. *Advanced Functional Materials 30* (4 2020), 1909910.

[50] Haneef, H. F., Zeidell, A. M., and Jurchescu, O. D. Charge carrier traps in organic semiconductors: a review on the underlying physics and impact on electronic devices. *Journal of Materials Chemistry C 8* (1 2020), 759–787.

[51] Hanke, T. Material science and engineering ontology.

[52] Hansen, K., Biegler, F., Ramakrishnan, R., Pronobis, W., Lilienfeld, O. A. V., Müller, K. R., and Tkatchenko, A. Machine learning predictions of molecular properties: Accurate many-body potentials and nonlocality in chemical space. *Journal of Physical Chemistry Letters 6* (6 2015), 2326–2331.

[53] Hansen, K., Montavon, G., Biegler, F., Fazli, S., Rupp, M., Scheffler, M., Lilienfeld, O. A. V., Tkatchenko, A., and Müller, K. R. Assessment and validation of machine learning methods for predicting molecular atomization energies. *Journal of Chemical Theory and Computation 9* (8 2013), 3404–3419.

[54] He, H., Bai, Y., Garcia, E. A., and Li, S. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. *Proceedings of the International Joint Conference on Neural Networks* (2008), 1322–1328.

[55] He, Z., Lin, D., Lau, T., and Wu, M. Gradient boosting machine: A survey.

[56] Hemingway, E. G., and O'Reilly, O. M. Perspectives on euler angle singularities, gimbal lock, and the orthogonality of applied forces and applied moments. *Multibody System Dynamics 44* (9 2018), 31–56.

[57] Hoerl, A. E., and Kennard, R. W. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics 42*, 1 (2000), 80–86.

[58] Hong, Y., Hou, B., Jiang, H., and Zhang, J. Machine learning and artificial neural network accelerated computational discoveries in materials science. *Wiley Interdisciplinary Reviews: Computational Molecular Science 10* (5 2020), e1450.

[59] Horowitz, G. Organic field-effect transistors. *Advanced Materials - Wiley Online Library* (1998).

[60] Horsch, M. T., Chiacchiera, S., Seaton, M. A., Todorov, I. T., Šindelka, K., Lísal, M., Andreon, B., Kaiser, E. B., Mogni, G., Goldbeck, G., Kunze, R., Summer, G., Fiseni, A., Brüning, H., Schiffels, P., and Cavalcanti, W. L. Ontologies for the virtual materials marketplace. *KI - Kunstliche Intelligenz 34* (9 2020), 423–428.

[61] Howard, I. A., Abzieher, T., Hossain, I. M., Eggers, H., Schackmar, F., Ternes, S., Richards, B. S., Lemmer, U., and Paetzold, U. W. Coated and printed perovskites for photovoltaic applications. *Advanced Materials 31* (6 2019), 1806702.

[62] Hunger, K., and Herbst, W. Pigments, organic. *Ullmann's Encyclopedia of Industrial Chemistry* (6 2000).

[63] Ingolfsson, H. I., Neale, C., Carpenter, T. S., Shrestha, R., Lopez, C. A., Tran, T. H., Oppelstrup, T., Bhatia, H., Stanton, L. G., Zhang, X., Sundram, S., Natale, F. D., Agarwal, A., Dharuman, G., Schumacher, S. I. K., Turbyville, T., Gulten, G., Van, Q. N., Goswami, D., Jean-Francois, F., Agamasu, C., Chen, D., Hettige, J. J., Travers, T., Sarkar, S., Surh, M. P., Yang, Y., Moody, A., Liu, S.,

van Essen, B. C., Voter, A. F., Ramanathan, A., Hengartner, N. W., Simanshu, D. K., Stephen, A. G., Bremer, P. T., Gnanakaran, S., Glosli, J. N., Lightstone, F. C., McCormick, F., Nissley, D. V., and Streitz, F. H. Machine learning-driven multiscale modeling reveals lipid-dependent dynamics of ras signaling proteins. *Proceedings of the National Academy of Sciences of the United States of America 119* (1 2022), e2113297119.

[64] Jacobsson, T. J., Hultqvist, A., García-Fernández, A., Anand, A., Al-Ashouri, A., Hagfeldt, A., Crovetto, A., Abate, A., Ricciardulli, A. G., Vijayan, A., Kulkarni, A., Anderson, A. Y., Darwich, B. P., Yang, B., Coles, B. L., Perini, C. A., Rehermann, C., Ramirez, D., Fairen-Jimenez, D., Girolamo, D. D., Jia, D., Avila, E., Juarez-Perez, E. J., Baumann, F., Mathies, F., González, G. S., Boschloo, G., Nasti, G., Paramasivam, G., Martínez-Denegri, G., Näsström, H., Michaels, H., Köbler, H., Wu, H., Benesperi, I., Dar, M. I., Pehlivan, I. B., Gould, I. E., Vagott, J. N., Dagar, J., Kettle, J., Yang, J., Li, J., Smith, J. A., Pascual, J., Jerónimo-Rendón, J. J., Montoya, J. F., Correa-Baena, J. P., Qiu, J., Wang, J., Sveinbjörnsson, K., Hirselandt, K., Dey, K., Frohna, K., Mathies, L., Castriotta, L. A., Aldamasy, M. H., Vasquez-Montoya, M., Ruiz-Preciado, M. A., Flatken, M. A., Khenkin, M. V., Grischek, M., Kedia, M., Saliba, M., Anaya, M., Veldhoen, M., Arora, N., Shargaieva, O., Maus, O., Game, O. S., Yudilevich, O., Fassl, P., Zhou, Q., Betancur, R., Munir, R., Patidar, R., Stranks, S. D., Alam, S., Kar, S., Unold, T., Abzieher, T., Edvinsson, T., David, T. W., Paetzold, U. W., Zia, W., Fu, W., Zuo, W., Schröder, V. R., Tress, W., Zhang, X., Chiang, Y. H., Iqbal, Z., Xie, Z., and Unger, E. An open-access database and analysis tool for perovskite solar cells based on the fair data principles. *Nature*

Energy 2021 7:1 7 (12 2021), 107–115.

[65] Jain, A., Ong, S. P., Hautier, G., Chen, W., Richards, W. D., Dacek, S., Cholia, S., Gunter, D., Skinner, D., Ceder, G., and Persson, K. A. Commentary: The materials project: A materials genome approach to accelerating materials innovation. *APL Materials 1* (7 2013), 011002.

[66] Jang, J., Nam, S., Im, K., Hur, J., Cha, S. N., Kim, J., Son, H. B., Suh, H., Loth, M. A., Anthony, J. E., Park, J. J., Park, C. E., Kim, J. M., and Kim, K. Highly crystalline soluble acene crystal arrays for organic transistors: Mechanism of crystal growth during dip-coating. *Advanced Functional Materials 22* (3 2012), 1005–1014.

[67] Jonathan, L., Diguna, L. J., Samy, O., Muqoyyanah, M., Bakar, S. A., Birowosuto, M. D., and Moutaouakil, A. E. Hybrid organicndash;inorganic perovskite halide materials for photovoltaics towards their commercialization. *Polymers 2022, Vol. 14, Page 1059 14* (3 2022), 1059.

[68] Juarez-Perez, E. J., Hawash, Z., Raga, S. R., Ono, L. K., and Qi, Y. Thermal degradation of ch3nh3pbi3 perovskite into nh3 and ch3i gases observed by coupled thermogravimetry-mass spectrometry analysis. *Energy Environmental Science 9* (11 2016), 3406–3410.

[69] Juarez-Perez, E. J., Ono, L. K., Maeda, M., Jiang, Y., Hawash, Z., and Qi, Y. Photodecomposition and thermal decomposition in methylammonium halide lead perovskites and inferred design principles to increase photovoltaic device stability. *Journal of Materials Chemistry A 6* (5 2018), 9604–9612.

[70] Juarez-Perez, E. J., Ono, L. K., and Qi, Y. Thermal degradation of formamidinium based lead halide perovskites into sym-triazine

and hydrogen cyanide observed by coupled thermogravimetry-mass spectrometry analysis. *Journal of Materials Chemistry A 7* (7 2019), 16912–16919.

[71] Juarez-Perez, E. J., Ono, L. K., Uriarte, I., Cocinero, E. J., and Qi, Y. Degradation mechanism and relative stability of methylammonium halide based perovskites analyzed on the basis of acid-base theory. *ACS Applied Materials and Interfaces 11* (4 2019), 12586–12593.

[72] Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Å½ídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., Hassabis, D., and Hassabis, D. Highly accurate protein structure prediction with alphafold. *Nature 596* (2021), 583.

[73] Jung, Y. S., Hwang, K., Heo, Y. J., Kim, J. E., Vak, D., and Kim, D. Y. Progress in scalable coating and roll-to-roll compatible printing processes of perovskite solar cells toward realization of commercialization. *Advanced Optical Materials 6* (5 2018), 1701182.

[74] Kang, J., and Wang, L. W. High defect tolerance in lead halide perovskite cspbbr3. *Journal of Physical Chemistry Letters 8* (1 2017), 489–493.

[75] Kantareddy, S. N. R., Mathews, I., Sun, S., Layurova, M., Thapa, J., Correa-Baena, J.-P., Buonassisi, R. B. T., Sarma, S. E., and Peters, I. M. Perovskite pv-powered rfid: enabling

low-cost self-powered iot sensors. *IEEE Sensors Journal 20* (9 2019), 471–478.

[76] Ke, J. C.-R., Walton, A. S., Lewis, D. J., Tedstone, A., O'Brien, P., Thomas, A. G., and Flavell, W. R. In situ investigation of degradation at organometal halide perovskite surfaces by x-ray photoelectron spectroscopy at realistic water vapour pressure. *Chemical Communications 53* (5 2017), 5231–5234.

[77] Khatib, M. E., and Jong, W. A. D. Ml4chem: A machine learning package for chemistry and materials science.

[78] Kim, C. H., Hlaing, H., Payne, M. M., Yager, K. G., Bonnassieux, Y., Horowitz, G., Anthony, J. E., and Kymissis, I. Strongly correlated alignment of fluorinated 5,11-bis(triethylgermylethynyl)anthradithiophene crystallites in solution-processed field-effect transistors. *ChemPhysChem 15* (10 2014), 2913–2916.

[79] Kim, S. J., and Lee, J. S. Flexible organic transistor memory devices. *Nano Letters 10* (8 2010), 2884–2890.

[80] Kojima, A., Teshima, K., Shirai, Y., and Miyasaka, T. Organometal halide perovskites as visible-light sensitizers for photovoltaic cells. *Journal of the American Chemical Society 131* (5 2009), 6050–6051.

[81] Kordt, P., Holst, J. J. V. D., Helwi, M. A., Kowalsky, W., May, F., Badinski, A., Lennartz, C., and Andrienko, D. Modeling of organic light emitting diodes: From molecular to device properties. *Advanced Functional Materials 25* (4 2015), 1955–1971.

[82] Koutsiaki, C., Kaimakamis, T., Zachariadis, A., Papamichail, A., Kamaraki, C., Fachouri, S., Gravalidis, C., Laskarakis, A., and Logothetidis, S. Efficient combination of

roll-to-roll compatible techniques towards the large area deposition of a polymer dielectric film and the solution-processing of an organic semiconductor for the field-effect transistors fabrication on plastic substrate. *Organic Electronics 73* (10 2019), 231–239.

[83] Kühne, T. D., Iannuzzi, M., Ben, M. D., Rybkin, V. V., Seewald, P., Stein, F., Laino, T., Khaliullin, R. Z., Schütt, O., Schiffmann, F., Golze, D., Wilhelm, J., Chulkov, S., Bani-Hashemian, M. H., Weber, V., Borštnik, U., Taillefumier, M., Jakobovits, A. S., Lazzaro, A., Pabst, H., Müller, T., Schade, R., Guidon, M., Andermatt, S., Holmberg, N., Schenter, G. K., Hehn, A., Bussy, A., Belleflamme, F., Tabacchi, G., Glöss, A., Lass, M., Bethune, I., Mundy, C. J., Plessl, C., Watkins, M., VandeVondele, J., Krack, M., and Hutter, J. Cp2k: An electronic structure and molecular dynamics software package - quickstep: Efficient and accurate electronic structure calculations. *The Journal of Chemical Physics 152* (5 2020), 194103.

[84] Kymissis, I. Organic field effect transistors.

[85] Lamport, Z. A., Barth, K. J., Lee, H., Gann, E., Engmann, S., Chen, H., Guthold, M., McCulloch, I., Anthony, J. E., Richter, L. J., DeLongchamp, D. M., and Jurchescu, O. D. A simple and robust approach to reducing contact resistance in organic transistors. *Nature Communications 2018 9:1 9* (12 2018), 1–8.

[86] Lamport, Z. A., Haneef, H. F., Anand, S., Waldrip, M., and Jurchescu, O. D. Tutorial: Organic field-effect transistors: Materials, structure and operation. *Journal of Applied Physics 124* (8 2018), 071101.

[87] Li, D., Liao, P., Shai, X., Huang, W., Liu, S., Li, H., Shen, Y., and Wang, M. Recent progress on stability issues of organic-

inorganic hybrid lead perovskite-based solar cells. *RSC Advances 6* (9 2016), 89356–89366.

[88] Li, H., Armiento, R., and Lambrix, P. An ontology for the materials design domain. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 12507 LNCS* (2020), 212–227.

[89] Li, Z., Klein, T. R., Kim, D. H., Yang, M., Berry, J. J., Hest, M. F. V., and Zhu, K. Scalable fabrication of perovskite solar cells. *Nature Reviews Materials 2018 3:4 3* (3 2018), 1–20.

[90] Liu, H., Xu, J., Li, Y., and Li, Y. Aggregate nanostructures of organic molecular materials. *Accounts of Chemical Research 43* (12 2010), 1496–1508.

[91] Liz-Marzán, L. M., and Kamat, P. V. Nanoscale materials. *Nanoscale Materials* (2004), 1–3.

[92] Lloyd, S. P. Least squares quantization in pcm. *IEEE TRANSACTIONS ON INFORMATION THEORY 28* (1982).

[93] Lo, Y. C., Rensi, S. E., Torng, W., and Altman, R. B. Machine learning in chemoinformatics and drug discovery. *Drug Discovery Today 23* (8 2018), 1538–1546.

[94] Lomax, S. Q. Phthalocyanine and quinacridone pigments: their history, properties and use. *Studies in Conservation 50* (6 2013), 19–29.

[95] Long, J., Huang, Z., Zhang, J., Hu, X., Tan, L., and Chen, Y. Flexible perovskite solar cells: device design and perspective. *Flexible and Printed Electronics 5* (2 2020), 013002.

[96] Lorenzoni, A., Baldoni, M., Besley, E., and Mercuri, F. Noncovalent passivation of supported phosphorene for device applica-

tions: from morphology to electronic properties. *Physical Chemistry Chemical Physics 22* (6 2020), 12482–12488.

[97] Lorenzoni, A., Conte, A. M., Pecchia, A., and Mercuri, F. Nanoscale morphology and electronic coupling at the interface between indium tin oxide and organic molecular materials. *Nanoscale 10* (5 2018), 9376–9385.

[98] Lorenzoni, A., Gallino, F., Muccini, M., and Mercuri, F. Theoretical insights on morphology and charge transport properties of two-dimensional n,n'-ditridecylperylene-3,4,9,10-tetra carboxylic diimide aggregates. *RSC Advances 6* (4 2016), 40724–40730.

[99] Lorenzoni, A., Muccini, M., and Mercuri, F. Correlation between gate-dielectric morphology at the nanoscale and charge transport properties in organic field-effect transistors. *RSC Advances 5* (1 2015), 11797–11805.

[100] Lorenzoni, A., Muccini, M., and Mercuri, F. Morphology and electronic properties of n,n'-ditridecylperylene-3,4,9,10-tetracarboxylic diimide layered aggregates: From structural predictions to charge transport. *Journal of Physical Chemistry C 121* (10 2017), 21857–21864.

[101] Lorenzoni, A., Muccini, M., and Mercuri, F. A computational predictive approach for controlling the morphology of functional molecular aggregates on substrates. *Advanced Theory and Simulations 2* (12 2019), 1900156.

[102] Lv, T., Yan, P., and He, W. Survey on json data modelling. *Journal of Physics: Conference Series 1069* (8 2018), 012101.

[103] MacKenzie, R. C., Kirchartz, T., Dibb, G. F., and Nelson, J. Modeling nongeminate recombination in p3ht:pcbm solar cells. *Journal of Physical Chemistry C 115* (5 2011), 9806–9813.

[104] MacKenzie, R. C., Shuttle, C. G., Chabinyc, M. L., and Nelson, J. Extracting microscopic device parameters from transient photocurrent measurements of p3ht:pcbm solar cells. *Advanced Energy Materials 2* (6 2012), 662–669.

[105] Macqueen, J. Some methods for classification and analysis of multivariate observations. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability 1* (1967), 281–297.

[106] Manser, J. S., Christians, J. A., and Kamat, P. V. Intriguing optoelectronic properties of metal halide perovskites. *Chemical Reviews 116* (11 2016), 12956–13008.

[107] Markowich, P. A. The stationary semiconductor device equations.

[108] Matteocci, F., Cinà, L., Lamanna, E., Cacovich, S., Divitini, G., Midgley, P. A., Ducati, C., and Carlo, A. D. Encapsulation for long-term stability enhancement of perovskite solar cells. *Nano Energy 30* (12 2016), 162–172.

[109] McInnes, L., Healy, J., and Melville, J. Umap: Uniform manifold approximation and projection for dimension reduction.

[110] Mei, Y., Loth, M. A., Payne, M., Zhang, W., Smith, J., Day, C. S., Parkin, S. R., Heeney, M., McCulloch, I., Anthopoulos, T. D., Anthony, J. E., and Jurchescu, O. D. High mobility field-effect transistors with versatile processing from a small-molecule organic semiconductor. *Advanced Materials 25* (8 2013), 4352–4357.

[111] Meller, J. J. Molecular dynamics. *Encyclopedia of Life Sciences* (2001).

[112] Mercuri, F. Semiempirical calculations on the electronic properties of finite-length models of carbon nanotubes based on clar sextet theory. *Molecular Simulation 34* (2008), 905–908.

[113] Mercuri, F. Theoretical investigations on the healing of monovacancies in single-walled carbon nanotubes by adsorption of carbon monoxide. *Journal of Physical Chemistry C 114* (12 2010), 21322–21326.

[114] Mercuri, F., Re, N., and Sgamellotti, A. Influence of substituents and length of silanylene units on the electronic structure of π-conjugated polymeric organosilicon systems. *Journal of Molecular Structure: THEOCHEM 489* (10 1999), 35–41.

[115] Mercuri, F., and Sgamellotti, A. Functionalization of carbon nanotubes with vaska's complex: A theoretical approach. *Journal of Physical Chemistry B 110* (8 2006), 15291–15294.

[116] Miller, A., and Abrahams, E. Impurity conduction at low concentrations. *Physical Review 120* (11 1960), 745.

[117] Min, H., Lee, D. Y., Kim, J., Kim, G., Lee, K. S., Kim, J., Paik, M. J., Kim, Y. K., Kim, K. S., Kim, M. G., Shin, T. J., and Seok, S. I. Perovskite solar cells with atomically coherent interlayers on sno2 electrodes. *Nature 2021 598:7881 598* (10 2021), 444–450.

[118] Mizoguchi, R., Vanwelkenhuysen, J., and Ikeda, M. (pdf) task ontology for reuse of problem solving knowledge, 1995.

[119] Mott, N. F. On the transition to metallic conduction in semiconductors. *https://doi.org/10.1139/p56-151 34* (12 2011), 1356–1368.

[120] Mueller, T., Kusne, A. G., and Ramprasad, R. Machine learning in materials science. *Reviews in Computational Chemistry 29* (5 2016), 186–273.

[121] Musen, M. A. The protégé project: a look back and a look forward. *AI Matters 1* (6 2015), 4–12.

[122] Natali, D., and Caironi, M. Charge injection in solution-processed organic field-effect transistors: Physics, models and characterization methods. *Advanced Materials 24* (3 2012), 1357–1387.

[123] Noy, N. F., and Mcguinness, D. L. Ontology development 101: A guide to creating your first ontology.

[124] O'Connor, D., and Hou, D. Manage the environmental risks of perovskites. *One Earth 4* (11 2021), 1534–1537.

[125] Ong, S. P. Accelerating materials science with high-throughput computations and machine learning. *Computational Materials Science 161* (2019), 143–150.

[126] Organization, W. H. Preventing disease through healthy environments: exposure to lead: a major public health concern. Technical documents, 2019.

[127] Panidi, J., Paterson, A. F., Khim, D., Fei, Z., Han, Y., Tsetseris, L., Vourlias, G., Patsalas, P. A., Heeney, M., and Anthopoulos, T. D. Remarkable enhancement of the hole mobility in several organic small-molecules, polymers, and small-molecule:polymer blend transistors by simple admixing of the lewis acid p-dopant b(c6f5)3. *Advanced Science 5* (1 2018).

[128] Patel, A., and Jain, S. Present and future of semantic web technologies: a research statement. *International Journal of Computers and Applications 43*, 5 (2021), 413–422.

[129] Patel, R. K., Pradhan, M. K., Hayajneh, M. T., Almomani, M. A., Al, H. B., hmoud, Sun, H., Wang, Q., Qian, J., Yin, Y., Shi, Y., and Li, Y. Unidirectional coating technology for organic field-effect transistors: materials and methods. *Semiconductor Science and Technology 30* (4 2015), 054001.

[130] Paterson, A. F., Singh, S., Fallon, K. J., Hodsden, T., Han, Y., Schroeder, B. C., Bronstein, H., Heeney, M., McCulloch, I., Anthopoulos, T. D., Paterson, A. F., McCulloch, I., Anthopoulos, T. D., Singh, S., Schroeder, B. C., Fallon, K. J., Bronstein, H., Hodsden, T., Han, Y., and Heeney, M. Recent progress in high-mobility organic transistors: A reality check. *Advanced Materials 30* (9 2018), 1801079.

[131] Payne, M. M., Parkin, S. R., Anthony, J. E., Kuo, C. C., and Jackson, T. N. Organic field-effect transistors from solution-deposited functionalized acenes with mobilities as high as 1 cm2/v·s. *Journal of the American Chemical Society 127* (4 2005), 4986–4987.

[132] Piane, F. L., Baldoni, M., Gaspari, M., and Mercuri, F. Mambo: a lightweight ontology for multiscale materials and applications.

[133] Pinheiro, G. A., Mucelini, J., Soares, M. D., Prati, R. C., Silva, J. L. D., and Quiles, M. G. Machine learning prediction of nine molecular properties based on the smiles representation of the qm9 quantum-chemistry dataset. *Journal of Physical Chemistry A 124* (11 2020), 9854–9866.

[134] Pippig, M., and Mercuri, F. Efficient evaluation of coulomb interactions in kinetic monte carlo simulations of charge transport. *The Journal of Chemical Physics 152* (4 2020), 164102.

[135] Pozo, F. G. D., Fabiano, S., Pfattner, R., Georgakopoulos, S., Galindo, S., Liu, X., Braun, S., Fahlman, M., Veciana, J., Rovira, C., Crispin, X., Berggren, M., and Mas-Torrent, M. Single crystal-like performance in solution-coated thin-film organic field-effect transistors. *Advanced Functional Materials 26* (4 2016), 2379–2386.

[136] Prud'hommeaux, E., and Seaborne, A. Sparql query language for rdf.

[137] Qi, B., and Wang, J. Fill factor in organic solar cells. *Physical Chemistry Chemical Physics 15* (5 2013), 8972–8982.

[138] Quin, L. Extensible markup language (xml).

[139] Quiroz, C. O. R., Shen, Y., Salvador, M., Forberich, K., Schrenker, N., Spyropoulos, G. D., Heumüller, T., Wilkinson, B., Kirchartz, T., Spiecker, E., Verlinden, P. J., Zhang, X., Green, M. A., Ho-Baillie, A., and Brabec, C. J. Balancing electrical and optical losses for efficient 4-terminal si-perovskite solar cells with solution processed percolation electrodes. *Journal of Materials Chemistry A 6* (2 2018), 3583–3592.

[140] Razza, S., Castro-Hermosa, S., Carlo, A. D., and Brown, T. M. Research update: Large-area deposition, coating, printing, and processing techniques for the upscaling of perovskite solar cell technology. *APL Materials 4* (9 2016), 091508.

[141] Reiser, P., Neubert, M., Eberhard, A., Torresi, L., Zhou, C., Shao, C., Metni, H., van Hoesel, C., Schopmans, H., Sommer, T., and Friederich, P. Graph neural networks for materials science and chemistry. *Communications Materials 3*, 1 (Nov 2022), 93.

[142] Ribeiro, R. P., Pfahringer, B., Branco, P., and Torgo, L. Smote for regression.

[143] Röhr, J. A., and Mackenzie, R. C. Analytical description of mixed ohmic and space-charge-limited conduction in single-carrier devices. *Journal of Applied Physics 128* (10 2020), 165701.

[144] Rolston, N., Scheideler, W. J., Flick, A. C., Chen, J. P., Elmaraghi, H., Sleugh, A., Zhao, O., Woodhouse, M., and

Dauskardt, R. H. Rapid open-air fabrication of perovskite solar modules. *Joule 4* (12 2020), 2675–2692.

[145] Rolston, N., Watson, B. L., Bailie, C. D., McGehee, M. D., Bastos, J. P., Gehlhaar, R., Kim, J. E., Vak, D., Mallajosyula, A. T., Gupta, G., Mohite, A. D., and Dauskardt, R. H. Mechanical integrity of solution-processed perovskite solar cells. *Extreme Mechanics Letters 9* (12 2016), 353–358.

[146] Rudnicki, R. An overview of the common core ontologies.

[147] Sahu, H., Yang, F., Ye, X., Ma, J., Fang, W., and Ma, H. Designing promising molecules for organic solar cells via machine learning assisted virtual screening. *Journal of Materials Chemistry A 7* (7 2019), 17480–17488.

[148] Saidaminov, M. I., Abdelhady, A. L., Murali, B., Alarousu, E., Burlakov, V. M., Peng, W., Dursun, I., Wang, L., He, Y., MacUlan, G., Goriely, A., Wu, T., Mohammed, O. F., and Bakr, O. M. High-quality bulk hybrid perovskite single crystals within minutes by inverse temperature crystallization. *Nature Communications 2015 6:1 6* (7 2015), 1–6.

[149] Sanchez-Lengeling, B., and Aspuru-Guzik, A. Inverse molecular design using machine learning: Generative models for matter engineering. *Science 361* (7 2018), 360–365.

[150] Schmidt, J., Marques, M. R., Botti, S., and Marques, M. A. Recent advances and applications of machine learning in solid-state materials science. *npj Computational Materials 2019 5:1 5* (8 2019), 1–36.

[151] Schütt, K. T., Sauceda, H. E., Kindermans, P.-J., Tkatchenko, A., and Müller, K.-R. SchNet - A deep learn-

ing architecture for molecules and materials. *The Journal of Chemical Physics 148*, 24 (03 2018). 241722.

[152] Schwartz, B. J. Conjugated polymers as molecular materials: How chain conformation and film morphology influence energy transfer and interchain interactions. *Annual Review of Physical Chemistry 54* (11 2003), 141–172.

[153] Sedgwick, P. Pearson's correlation coefficient. *BMJ 345* (7 2012).

[154] Selberherr, S. Analysis and simulation of semiconductor devices. *Analysis and Simulation of Semiconductor Devices* (1984).

[155] Selli, D., Baldoni, M., Sgamellotti, A., and Mercuri, F. Redox-switchable devices based on functionalized graphene nanoribbons. *Nanoscale 4* (2 2012), 1350–1354.

[156] Sirringhaus, H. Device physics of solution-processed organic field-effect transistors. *Advanced Materials 17* (10 2005), 2411–2425.

[157] Snaith, H. J. Perovskites: The emergence of a new era for low-cost, high-efficiency solar cells. *Journal of Physical Chemistry Letters 4* (11 2013), 3623–3630.

[158] Snaith, H. J., Abate, A., Ball, J. M., Eperon, G. E., Leijtens, T., Noel, N. K., Stranks, S. D., Wang, J. T. W., Wojciechowski, K., and Zhang, W. Anomalous hysteresis in perovskite solar cells. *Journal of Physical Chemistry Letters 5* (5 2014), 1511–1515.

[159] Society, A. C. How a solar cell works.

[160] Søndergaard, R. R., Hösel, M., and Krebs, F. C. Roll-to-roll fabrication of large area functional organic materials. *Journal of Polymer Science Part B: Polymer Physics 51* (1 2013), 16–34.

[161] Spivak, D. I. Metric realization of fuzzy simplicial sets.

[162] Studer, R., Benjamins, V. R., and Fensel, D. Knowledge engineering: Principles and methods. *Data Knowledge Engineering 25* (3 1998), 161–197.

[163] Supervisor, R. P. A. R., and Torgo, L. Utility-based regression.

[164] Temiño, I., Pozo, F. G. D., Ajayakumar, M. R., Galindo, S., Puigdollers, J., and Mas-Torrent, M. A rapid, low-cost, and scalable technique for printing state-of-the-art organic field-effect transistors. *Advanced Materials Technologies 1* (8 2016), 1600090.

[165] Tessler, N., Preezant, Y., Rappaport, N., and Roichman, Y. Charge transport in disordered organic materials and its relevance to thin-film devices: A tutorial review. *Advanced Materials 21* (7 2009), 2741–2761.

[166] Thompson, A. P., Aktulga, H. M., Berger, R., Bolintineanu, D. S., Brown, W. M., Crozier, P. S., in 't Veld, P. J., Kohlmeyer, A., Moore, S. G., Nguyen, T. D., Shan, R., Stevens, M. J., Tranchida, J., Trott, C., and Plimpton, S. J. Lammps - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Computer Physics Communications 271* (2 2022), 108171.

[167] Unger, E. L., Hoke, E. T., Bailie, C. D., Nguyen, W. H., Bowring, A. R., Heumüller, T., Christoforo, M. G., and McGehee, M. D. Hysteresis and transient behavior in current-voltage measurements of hybrid-perovskite absorber solar cells. *Energy Environmental Science 7* (10 2014), 3690–3698.

[168] Volpi, R. Charge transport simulations for organic electronics: A kinetic monte carlo approach.

[169] Vu, K., Snyder, J., Li, L., Rupp, M., Chen, B. F., Khelif, T., MÃ¼LLER, K.-R., and Burke, K. Understanding kernel ridge regression: Common behaviors from simple functions to density functionals.

[170] Vvedensky, D. D. Multiscale modelling of nanostructures. *J. Phys.: Condens. Matter 16* (2004), 1537–1576.

[171] Walker, C., and Alrehamy, H. Personal data lake with data gravity pull. *Proceedings - 2015 IEEE 5th International Conference on Big Data and Cloud Computing, BDCloud 2015* (10 2015), 160–167.

[172] Walschap, G. Metric structures in differential geometry.

[173] Wang, C., Zhang, Z., and Wang, Y. Quinacridone-based π-conjugated electronic materials. *Journal of Materials Chemistry C 4* (10 2016), 9918–9936.

[174] Warta, W., and Karl, N. Hot holes in naphthalene: High, electric-field-dependent mobilities. *Physical Review B 32* (7 1985), 1172.

[175] Xie, T., and Grossman, J. C. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Phys. Rev. Lett. 120* (Apr 2018), 145301.

[176] Xu, Y., Liu, C., Khim, D., and Noh, Y. Y. Development of high-performance printed organic field-effect transistors and integrated circuits. *Physical Chemistry Chemical Physics 17* (10 2015), 26553–26574.

[177] Yang, S., Park, S., Bmtinger, J., Bonnassieux, Y., and Kymissista, L. P-99: Pneumatic nozzle printing as a versatile approach to crystal growth management and patterning of printed organic thin film transistors. *SID Symposium Digest of Technical Papers 47* (5 2016), 1502–1505.

[178] Yershova, A., Jain, S., Lavalle, S. M., and Mitchell, J. C. Generating uniform incremental grids on so(3) using the hopf fibration. *The International Journal of Robotics Research 29* (11 2009), 801–812.

[179] Yin, W. J., Shi, T., and Yan, Y. Unique properties of halide perovskites as possible origins of the superior solar cell performance. *Advanced Materials 26* (7 2014), 4653–4658.

[180] Ying, C., Cai, T., Luo, S., Zheng, S., Ke, G., He, D., Shen, Y., and Liu, T.-Y. Do transformers really perform bad for graph representation? In *Neural Information Processing Systems* (2021).

[181] Yuan, Y., Giri, G., Ayzner, A. L., Zoombelt, A. P., Mannsfeld, S. C., Chen, J., Nordlund, D., Toney, M. F., Huang, J., and Bao, Z. Ultra-high mobility transparent organic thin film transistors grown by an off-centre spin-coating method. *Nature Communications 2014 5:1 5* (1 2014), 1–9.

[182] Yuan, Y., Wang, Q., Shao, Y., Lu, H., Li, T., Gruverman, A., and Huang, J. Electric-field-driven reversible conversion between methylammonium lead triiodide perovskites and lead iodide at elevated temperatures. *Advanced Energy Materials 6* (1 2016), 1501803.

[183] Zeng, X., Xiang, H., Yu, L., Wang, J., Li, K., Nussinov, R., and Cheng, F. Accurate prediction of molecular properties and drug targets using a self-supervised image representation learning framework. *Nature Machine Intelligence 2022 4:11 4* (11 2022), 1004–1016.

[184] Zhang, S., Liu, Y., and Xie, L. Molecular mechanics-driven graph neural network with multiplex graph for molecular structures.

[185] Zhao, Y., Li, J., Liao, C., and Shen, X. Bridging the gap between deep learning and sparse matrix format selection. *ACM SIGPLAN Notices 53* (2 2018), 94–108.

[186] Zidan, H., and Abu-Elnader, M. Structural and optical properties of pure pmma and metal chloride-doped pmma films. *Physica B: Condensed Matter 355*, 1 (2005), 308–317.