Technische Universität München
TUM School of Natural Sciences

Machine Learning Based Screening of Double Perovskites for Photovoltaic Applications

Elisabetta Landini

Vollständiger Abdruck der von der TUM School of Natural Sciences der Technischen Universität München zur Erlangung des akademischen Grades einer

Doktorin der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitz: Priv.-Doz. Dr. Martin Tschurl

Prüfer*innen der Dissertation:
1. Prof. Dr. Karsten Reuter
2. Prof. Dr. David Egger

Die Dissertation wurde am 07.06.2023 bei der Technischen Universität München eingereicht und durch die TUM School of Natural Sciences am 01.08.2023 angenommen.

Abstract

Materials based on the perovskite crystal structure and its derivatives find many different applications in materials science, thanks to their compositional diversity and wide variety of chemical and physical properties. In the context of materials discovery, perovskite compounds with their fixed crystal structure represent ideal candidates to study the interplay between chemical composition and properties.

In this work we focus on the double perovskite structure ABC_2D_6, and the properties that can be obtained with different combinations of elements in the four lattice sites. In particular, we start by screening lead-free halide double perovskites to select ideal candidates for photovoltaic applications using as criteria their ability to absorb light in the visible range, their thermodynamical stability, their power conversion efficiency and the mobility of the carriers. The first step of the screening process was accelerated by machine learning via the use of a convolutional neural network to map the composition to the band gap of the materials. High throughput screening of materials as in this case, relies on the possibility of fully enumerating the search space and calculating the target property for all the elements in the space, while choosing a compromise between accuracy and computational cost.

This kind of neural network-based regression model, however, does not explicitly highlight the contribution of each element to the predicted property. To gain more insight in the interaction between elements placed at different sites in the double perovskite structure, and their relation to the band gap, we chose a machine learning model with interpretable parameters, based on a message passing neural network with a self-attention mechanism. We used this model to predict atomic energy levels in the valence and conduction band of the materials and calculate the gap. The weights that the network places on each lattice site when occupied by different elements allow us to classify the perovskites in families with specific behaviors.

Another possible approach consists in the inverse design of materials starting from a desired property. For this task, deep generative models have proven to be powerful tools, especially when studying very large chemical spaces. Here we used a dataset of double perovskites to compare the performance of three generative models for the inverse design of materials with a given formation energy. We defined several metrics to assess the ability of the three models (a Variational Autoencoder, a Generative Adversarial Network and a Reinforcement Learning model) to generate compositions with the target property with high precision, while at the same time providing a high number of diverse candidates with the desired property.

Contents

1 Introduction

1.1 Perovskite solar cells

Perovskites are ternary compounds with chemical formula ABX_3, where A is a cation coordinated with 12 X anions, and B is a cation in octahedral coordination with 6 X anions (Figure 1.1). This kind of crystal structure can accommodate a vast range

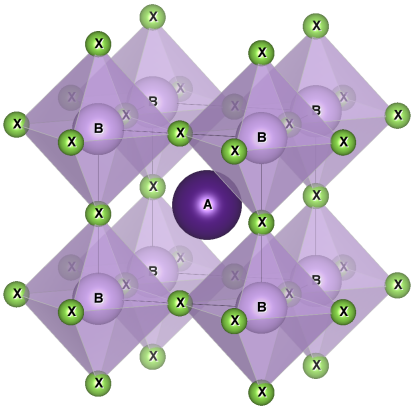


Figure 1.1 Perovskite structure ABX_3

of elements, leading to the formation of materials with a wide variety of properties. In particular, the A site can be occupied by large inorganic cations or small organic cations, opening the possibility to form hybrid organic/inorganic structures [1]. Two empirical geometric factors are commonly used to establish if a combination of elements can be stable in the perovskite structure, namely the octahedral factor

$$\mu = \frac{r_B}{r_X} \quad , \tag{1.1}$$

and the Goldschmidt tolerance factor [2]

$$t = \frac{r_A + r_X}{\sqrt{2}(r_B + r_X)} \quad , \tag{1.2}$$

where r indicates the ionic radius. Both parameters are based on a rigid spheres model, where the elements touch to form close packed structures and cavities. The octahedral factor is calculated by forming an octahedron of X atoms that touch each other, and measuring how large can the B atom be if it has to be inserted inside the central cavity. Empirically, it has been found that B and X can form an octahedral coordination when $0.44 < \mu < 0.90$ [3]. The Goldschmidt tolerance factor t indicates whether the A cation can fit in the cavity formed by the BX_6 octahedra. A perovskite is considered geometrically stable if $0.8 \leq t \leq 1$, with a perfect cubic structure if $t = 1$ and some distortion or tilting of the BX_6 octahedra when t is closer to 0.8. With values below 0.8 (A too small) or above 1 (A too large) the perovskite structure is unstable, but some perovskite-like phases might form, for example hexagonal structures ($\mu < 0.8$) or layered perovskites ($\mu > 1$).

Perovskites find applications in many fields, such as light emitting diodes [4–6], photocatalytic water splitting [7, 8], photodetectors [9, 10], fuel cells [11, 12], lasers [13, 14], and solar cells [15–19]. Perovskite solar cells belong to the so-called third generation solar cells, where the generations conventionally are [20]

- *First generation*: single junction solar cells based on Si (single crystal or polycristalline) and GaAs. These cells have high efficiency and long lifetime but also high production cost.

- *Second generation*: thin films based on CdTe, Copper Indium Gallium Selenide (CIGS), and amorphous Si. These are also based on single junctions, they have a lower production cost due to fewer material needed for the thin film, but also lower efficiency compared to the bulk material.

- *Third generation*: these cells are based on new materials with low manifacturing costs, such as organic semiconductors, polymers, dyes, and perovskites, with a single or multi junction architecture, with a thin film geometry or as quantum dots.

The first perovskite-based solar cell, a dye-sensitized solar cell containing Methylammonium Lead Iodide (MAPI) as organic sensitizer, was built in 2009 by Kojima et al. [21] and was reported to have a Power Conversion Efficiency (PCE) of 3.8%. However these structures showed very poor stability due to the dissolution of the perovskite in the iodide-based liquid electrolyte. This value increased to 9.7% [22] just a few years later when MAPI was used in a solid state solar cell. Today, the most commonly used hybrid organic-inorganic perovskite solar cells are based on MAPI, (MA)$PbBr_3$ [23] or Formamidinium Lead Iodide (FAPI) or the mixed perovskites (FA,MA)PbI_3 and (MA,FA)$Pb(I,Br)_3$ [1], and efficiencies as high as 25% can be achieved [24]. This would make them comparable in performance to commonly used materials such

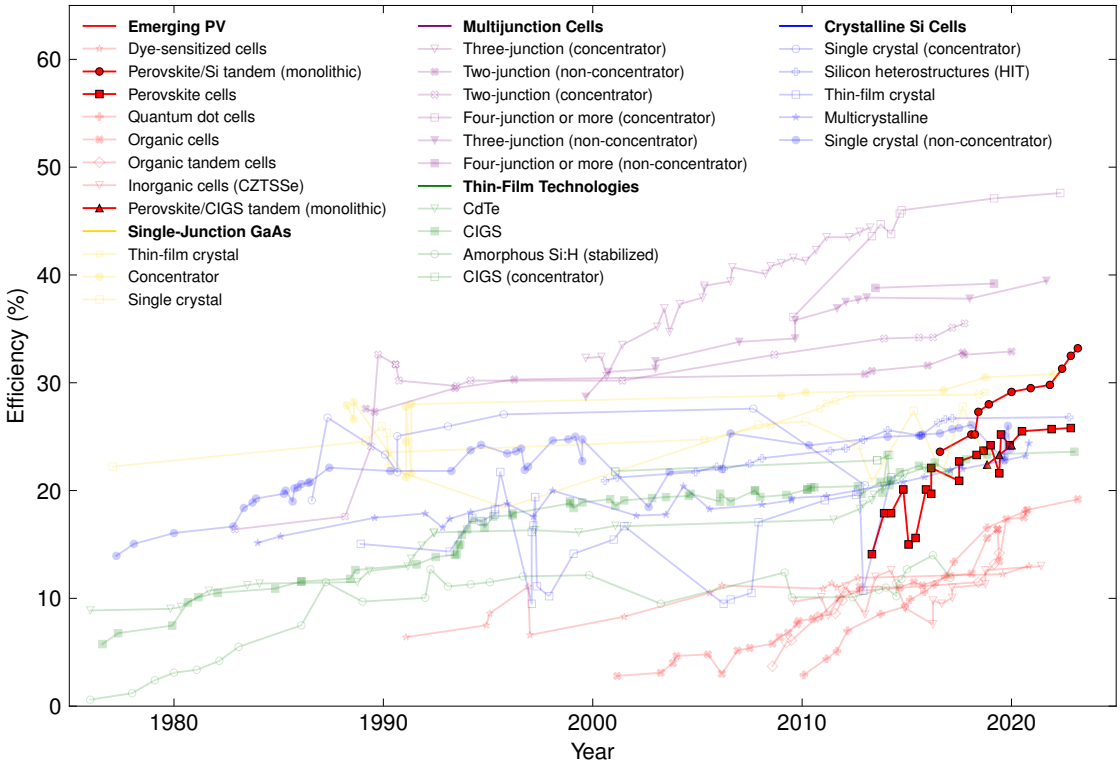


Figure 1.2 Best research-cell efficiency chart adapted from reference [25]

as CdTe, CIGS and polycrystalline Si. Furthermore, the components are easily available and the synthesis happens at low temperature, with a reduced production cost.

The solid state perovskite solar cells are composed of an absorber layer, an Electron Transport Layer (ETL), a Hole Transport Layer (HTL), a metal electrode and a transparent electrode. The absorber layer, i.e. the perovskite, absorbs light and creates electron-hole pairs. It is important for this layer to absorb photons in the visible range, around the peak of intensity of the solar spectrum. The generated electrons and holes are then separated and carried via the ETL and HTL [26], respectively, towards the electrodes and the external circuit. The HTL has the function of transferring an electron to the oxidized perovskite and transport the holes away from the interface to prevent carrier recombination, i.e. the annihilation of an electron and a hole to produce energy in the form of photon or phonons. The highest occupied state of the HTL must lie above the valence band of the perovskite to provide a driving force in the right direction, and the HTL material needs to have a high hole mobility. Additionally, it should have high photochemical stability, suitable solubility in organic solvent and form films of good quality. Some commonly used HTL materials are organic materials such as 2,2',7,7'-Tetrakis[N,N-di(4-methoxyphenyl)amino]-9,9'-spirobifluorene (spiro-OMeTAD) or poly(3,4-ethylene dioxythiophene)-poly(styrene sulfonate) (PEDOT:PSS), or inorganic materials such as graphene oxide, carbon, CuSCN, $CuZnSnS_2$, $CuInS_2$, CuI, Cu_2O, CuO, and NiO. The ETL transfers the photogenerated electrons to the electrode and blocks the holes at the interface with the perovskite. Also in this case the band alignment needs to allow the current to flow in the right direction, i.e. with the electrons moving from the high-energy conduction band of the perovskite to the lower energy unoccupied states of the ETL. The ETL also needs high stability, high electron mobility, and a good interface contact with the perovskite layer. Some widely used examples are conductive oxides such as TiO_2, SnO_2, and ZnO [27]. The metal electrode is usually Au or Al, while the conductive glass is Indium Tin Oxide (ITO) or Fluorine-doped Tin Oxide (FTO). The device can have two architectures: *n-i-p* (regular) if the ETL follows the conductive glass, and *p-i-n* (inverted) if the light hits the HTL first. In both cases the perovskite layer could be mesoporous or planar (Figure 1.3).

The outstanding performance of these materials is due to their high absorption coefficient, long diffusion length, and high and balanced carrier mobility of electrons and holes [28, 29].

From the point of view of the electronic structure, the presence of Pb^{2+} with a lone pair in the s shell is crucial. The Valence Band Maximum (VBM) shows contributions from the σ-antibonding state between the $6s$ orbital of Pb and $5p$ orbital of I, while the Conduction Band Minimum (CBM) is formed mostly by the Pb $6p$ with a smaller contribution from I $5s$ [31]. The states from the organic part lie far from the band

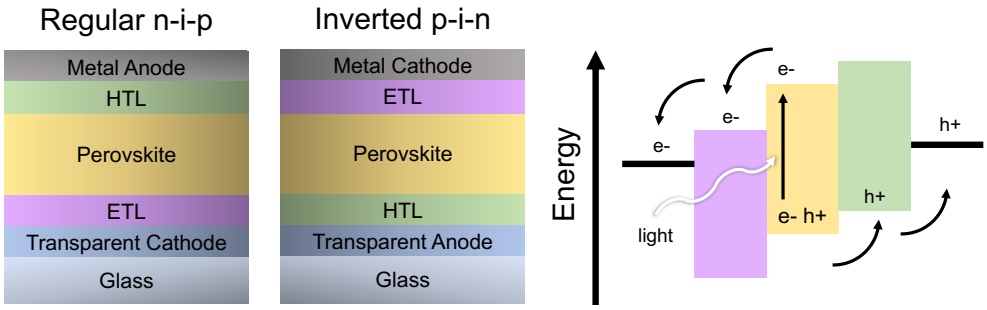


Figure 1.3 Device structure for a n-i-p and p-i-n architecture (left). Schematic band diagram for an n-i-p device (right)

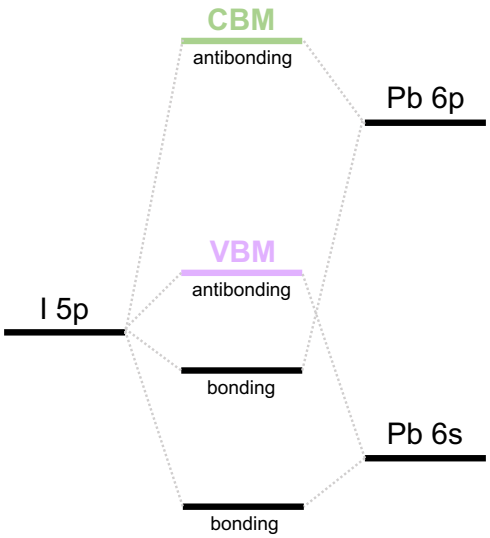


Figure 1.4 Orbital diagram of Pb and I in MAPI [30]

edges, so the role of the organic cation is mostly structural stabilization. The strong optical transition across the band gap has been attributed to the large contribution of Pb $6s$ and Pb $6p$ states to the band edges [32]. The measured direct gap of 1.6 eV [33] is also in an ideal range for visible light absorption. Theoretical studies that included Spin-Orbit Coupling (SOC) effects, revealed the presence of a splitting of the conduction band, leading to an indirect gap 60 meV below the direct gap [34]. This might contribute to the the observed low recombination rate and long lifetime of the minority carriers, but does not negatively affect the optical absorption.

The character of the band edges also contributes to bands with relatively high dispersion. In a nearly free electron approximation (2.2) the carriers are treated as free particles with kinetic energy $E_{kin} = \frac{\hbar^2 k^2}{2m^*}$, where m^* is a corrected mass, known as *effective mass*, that takes into account the interaction of the particle with the lattice. High dispersion, i.e. low curvature radius, indicates that the effective mass is low, and in MAPI as well as in many other semiconductors even lower than the electron rest mass. This in turn contributes to having carriers with high mobility. The low effective mass, combined with the high dielectric constant, also affects the excitonic properties of MAPI. An exciton is an excited electron-hole pair, bound together by an attractive Coulomb interaction between the two elementary charges, screened by the dielectric constant of the material ϵ. This bound pair can move through the crystal as a quasi-particle of neutral charge and reduced mass $\mu^* = m_e^* m_h^* / (m_e^* + m_h^*)$. In semiconductors, where the dielectric constant is usually large, excitons can extend over several unit cells (Mott-Wannier excitons). The binding energy E_B, i.e. the energy required to transform an exciton into two free carriers, can be calculated from a hydrogen-like model with the reduced mass and the screened Coulomb energy

$$E_B = \frac{\mu^*}{m_0} \frac{E_{Ry}}{\epsilon^2} \quad , \tag{1.3}$$

where $E_{Ry} = \frac{m_0 e^4}{8\epsilon_0^2 h^3 c} = 13.6\ eV$ is the Rydberg constant. The reported experimental values for the exciton binding energy in MAPI varies from 2 to 50 meV, suggesting that at least part of the excitons will dissociate in free carriers at room temperature [35–37].

An important factor that can reduce the maximum efficiency achievable by a solar cell is the loss of carriers via recombination. The recombination rate in a semiconductor with electron charge-carrier density n can be expressed as

$$\frac{\mathrm{d}n}{\mathrm{d}t} = -k_{SRH} \cdot n - k_{radiative} \cdot n^2 - k_{Auger} \cdot n^3 \quad . \tag{1.4}$$

The three constants control the rate of Shockley-Read-Hall (SRH) (non-radiative) trap-assisted recombination, band-to-band (radiative) recombination, and Auger (non-radiative) recombination. In perovskite solar cells at working conditions, it has been

shown that only a minor contribution comes from radiative and Auger recombination [1, 38]. The largest contribution derives from non-radiative recombination due to trap states deep in the band gap. In MAPI, among the point defects with low formation energy, only interstitial iodine and Pb vacancies form deep defect states. Moreover, the recombination of electrons trapped at positive interstitials involves a large lattice reorganization and therefore happens at a much lower rate than band-to-band transition. On the other hand, holes trapped at negative interstitals and Pb vacancies have a recombination rate comparable to the radiative recombination rate and contribute to decreasing the device efficiency [39]. Other sources of non-radiative recombination are the grain boundaries within the perovskite layer, and even more the interface between the perovskite and the ETL or HTL [40]. Overall, perovskite solar cells are less sensitive to defects than other inorganic semiconductors [1], and careful device engineering can effectively limit the non-radiative recombination loss.

However, these materials still face some challenges, due to the instability of the structure when exposed to humidity, light and heat, and the toxicity of Pb, that make encapsultaion in glass and epoxy necessary for commercialization. Furthermore, large scale application of these compounds, even encapsulated, is not ideal due to the presence of Pb that needs to be disposed of at the end of the life cycle of the cell. One possible route to solve this problem consists in substituting Pb^{2+} with Sn^{2+} or Ge^{2+}, because of their similar electronic structures.

The measured band gap of (MA)SnI_3 is 1.2–1.4 eV [41], which enables a broader absorption compared to MAPI [42]. However, the energy of the $5s$ orbital of Sn is higher than that of Pb $6s$, which makes the Sn-I bonds easy to break, leading to the formation of Sn vacancies and, in presence of oxygen and water, the formation of oxides and hydroxydes of Sn and and methylammonium iodide [43]. For this reason, also tin-based perovskite compounds need to be encapsulated.

Ge-based compounds such as $CsGeI_3$, (MA)GeI_3, and (FA)GeI_3 crystallize in a corner-sharing trigonal structure. These compounds present a direct band gap that ranges from 1.6 eV for $CsGeI_3$ to 2.35 eV for (FA)GeI_3, and strong absorption. However, similarly to thier Sn-based counterparts, they are highly unstable when exposed to air due to the oxidation of Ge^{2+} to Ge^{4+} [44, 45].

Further studies have shown that improved optical properties and stability are possible for mixed compounds such as (MA)$GeI_{3-y}Br_y$ [46] and (MA)$Sn_{1-x}Ge_xI_{3-y}Br_y$ [47].

1.2 Double Perovskites

Another strategy consists in substituting Pb-based single perovskites with double perovskites, with chemical formula ABC_2D_6. For solar cells, lead-free halide double perovskites have attracted significant attention. In halide double perovskites C a is monovalent cation, often alkali metal, A and B are two cations with charge +1 and +3 or

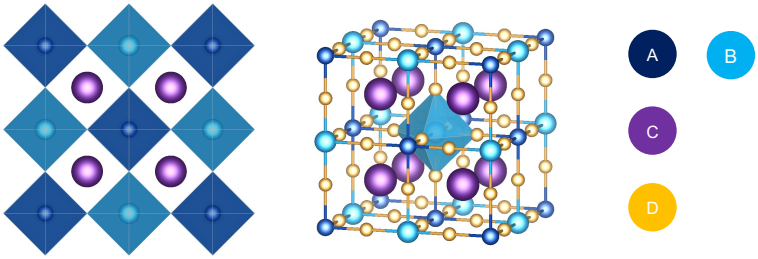


Figure 1.5 Double perovskite structure ABC_2D_6

+2 and +2, and D is a monovalent anion, usually a halide. The structure (Figure 1.5) is similar to that of a single perovskite but with alternated corner-sharing octahedra AD_6 and BD_6. The A cation is again coordinated to 12 D anions. Also in the case of double perovskites the stability can be empirically estimated via the octahedral and Goldschmidt tolerance factors, by using an average of the radii of A and B

$$\mu_{double\ perovskite} = \frac{\frac{1}{2}(r_A + r_B)}{r_D} \quad , \tag{1.5}$$

$$t_{double\ perovskite} = \frac{r_C + r_D}{\sqrt{2}\left[\frac{1}{2}(r_A + r_B) + r_D\right]} \quad . \tag{1.6}$$

Alternatively, a modified version of the Goldschmidt tolerance from Bartel et al. [48] can be used

$$\tau = \frac{r_D}{\frac{1}{2}(r_A + r_B)} - n_C \left\{ n_C - \frac{r_C / \left[\frac{1}{2}(r_A + r_B)\right]}{\ln\left\{r_C / \left[\frac{1}{2}(r_A + r_B)\right]\right\}} \right\} \quad , \tag{1.7}$$

where n_C is the oxidation state of C. This new tolerance factor was found with the Sure Independence Screening and Sparsifying Operator (SISSO) approach. The first term in τ contains the ratio between the D anion and the A/B cations, and has the same geometric interpretation of the octahedral factor. The term $\frac{r_C}{\left[\frac{1}{2}(r_A + r_B)\right]}$ indicates that when C and the mean of A and B have a similar size the perovskite structure, where site C and A/B have very different environments, is unfavourable respect to other structures where C and A/B have the same coordination. The oxidation state of C n_C increases the probability of forming a perovskite the more C is oxidized, but it also increases the importance of the radii ratio of C and A/B. A stable perovskite is expected when $\tau < 4.18$.

To imitate the favorable electronic properties of MAPI, another cation with a lone pair can be used. These include Tl^+ and Bi^{3+}, but Tl is known to be even more toxic

than lead, so only Bi is a feasible candidate. One of the most extensively studied halide double perovskites based on Bi is $AgBiCs_2Br_6$ [49–54]. This compound has a large (1.9-2.1 eV) and indirect gap, so light absorption is weak (Figure 1.6). The VBM has contributions from Ag $4d$ and Br $4p$, while the CBM is composed of states from Bi $6p$ and Br $4p$ antibonding states. The A cation doesn't contribute to the band edges. SOC effects are also very important in this material, due to the presence of

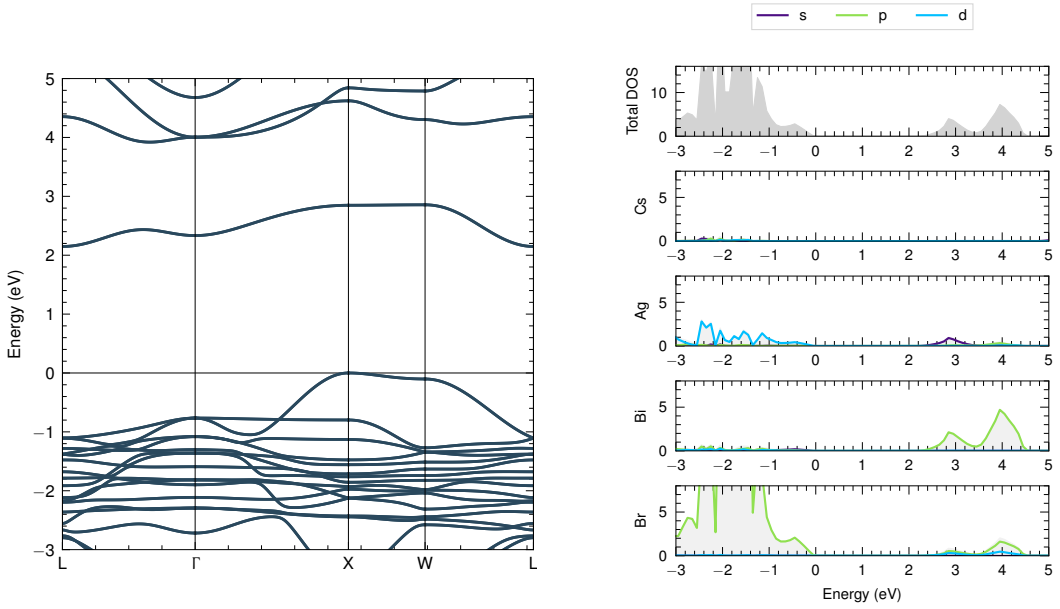


Figure 1.6 Band structure of $AgBiCs_2Br_6$ (left) and orbital-projected density of states (right). The (indirect) band gap is between the points Γ and L.

Bi states at the conduction band that experience a strong spin-orbit splitting [55]. The calculated effective masses along the three principal axes are $0.57m_e/0.57m_e/0.15m_e$ for the holes and $0.48m_e/0.28m_e/0.28m_e$ for the electrons, with an anisotropy ratio of 50% and 27% respectively. These values contribute to a good carrier mobility, despite the high anisotropy. The charge carrier lifetime in $AgBiCs_2Br_6$ is larger than that of lead-based halide single perovskites [56], which could allow the use of thicker films to improve the efficiency. Due to the limitations that come from the band gap, the efficiency of $AgBiCs_2Br_6$ never went beyond 6% [57]). A closely related compound is $AgInCs_2Cl_6$ [58, 59], where Bi^{3+} is substituted with In^{3+}. This material has a direct gap, however, the parity-forbidden transition [60] makes it a weak absorber. A small direct gap of 0.95 eV was found for $AgTlCs_2Br_6$ [61] but the high toxicity of Tl limits its application.

1.3 Band Gap Engineering

As mentioned before, the main limitation to the applicability of halide double perovskites to photovoltaics is their large indirect gap. In the following, we discuss possible strategies to modify in a controlled manner the band gap of a compound. The first strategy is by alloying, i.e. by substituting elements in an amount sufficient to have the impurities interact and form bands. The band gap of a disordered solid solution between ABC_2D_6 and $AB'C_2D_6$ can be estimated as [62]

$$E_G(AB_{1-x}B'_xC_2D_6) = (1-x)\cdot E_G(ABC_2D_6)+x\cdot E_G(AB'C_2D_6)+b\cdot x\cdot(1-x) \quad . \text{(1.8)}$$

The parameter b is an empirical factor known as bowing parameter. This is usually possible only if the cations B and B' have the same valency, such as in $AgBi_{1-x}Sb_xCs_2Br_6$ [63, 64]. In this case, the gap decreases with increasing x but remains indirect, because Sb^{3+} and Bi^{3+} have the same electronic structure but the filled $5s$ states of Sb^{3+} in the VB are higher in energy than the $6s$ states of Bi^{+}, due to relativistic effects. Similarly, the gap can be manipulated by alloying at the halide site, as in $AgBiCs_2Br_{6-x}I_x$ [65, 66], where the substitution of Br with I shifts the VBM at higher energies and the CBM at lower energies, effectively reducing the gap. When the oxidation state of the dopant element is different than the one in the host structure, some charged defects will be formed to maintain the overall charge neutrality of the structure. This generally limits the amount of dopant that can be introduced in the material to less than 1 atom % [67], but can additionally change the symmetry of the gap. When doping at the Bi site with In^{3+} or Tl^{3+}, as in $AgBi_{1-x}In_xCs_2Br_6$ [64] and $AgBi_{1-x}Tl_xCs_2Br_6$, a ns^2np^0 cation is substituted with a $(n-1)d^{10}ns^0$ cation. The empty s states of the dopant, together with the empty s states of Ag, form a band that shifts the CBM to Γ, while the removal of Bi lowers the VBM. Overall, the gap is reduced when increasing Tl doping, due to the dominant effect of lowering the CBM, and slightly increased when doping with In, due the the lowering of the VBM [67]. However, the gap remains indirect unless Bi is completely removed. The same effect was observed in $AgSb_{1-x}In_xCs_2Cl_6$ [68].

Other studies have shown that the application of a pressure on $AgBiCs_2Br_6$ of 15 GPa narrowed the gap from an initial value of 2.3 eV to 1.7 eV, and to 2 eV after the pressure was released [69]. Another factor that affects the band gap is the degree of disorder in the Ag-Bi arrangement [70]. This can be controlled via the growth temperature and speed [71].

1.4 High Throughput Screening of Double Perovskites

The large chemical space formed by (charge-balanced) single [72, 73] and double perovskites is potentially very large and has been frequently explored via first-principles based screening. In this context, a computational study by Volonakis et

al. [74] showed the possibility of tuning the gap by substituting different halides at the C site (Cl, Br, I), different monovalent cations at the A site (Cu, Ag, Au) and different trivalent cations at B (Bi, Sb). All the structures obtained had an indirect gap, but they suggested that a different A cation might induce an octahedral tilt and make the direct transition possible (i.e. not symmetry-forbidden). However, in this study the thermodynamic stability of said compounds was not considered. Filip et al. [75] evaluated the thermodynamical stability and the electronic properties of the compounds in the family $ABCs_2D_6$ (with A=Cu/Ag/Au, B = Sb/Bi, D = Cl/Br/I) via Density Functional Theory (DFT) using the PBE functional, and suggested that the solid solution $(Ag,Cu)BiCs_2Cl_6$ could be synthetized and have optimal properties for photovoltaic applications. Dai et al. [76] studied 18 In- and Ga-based double perovskites and found that four of these ($CuInCs_2Cl_6$, $AgInCs_2Br_6$, $CuGaCs_2Cl_6$, and $AgGaCs_2Br_6$) had a direct gap in the desirable 0.9-1.6 eV range. However only one ($AgInCs_2Br_6$) was found to be mechanically and thermodynamically stable. Zhao et al. [77] evaluated the structural and thermodynamical stability of 64 double perovskites (with C=Cs, A=Na/K/Rb/Cu/Ag/Au/In/Tl, B=Bi/Sb and D=F/Cl/Br/I) and proposed 11 non-toxic compounds with suitable band gaps and low carrier effective masses. In particular, $InSbCs_2Cl_6$ and $InBiCs_2Cl_6$ have promising direct bandgaps of 1.02 and 0.91 eV respectively. However, it has been shown [78] that In^+ can easily oxidate to In^{3+}, and therefore these materials are likely unstable. Volonakis et al. [79] showed that the instability of In^+ compounds decreases with increasing the size of the C cation, and that a solid solution such as $InBi(Cs,Ma,FA)_2Br_6$ would be thermodynamically stable and have a performance comparable to that of MAPI. Roknuzzaman et al. [80] studied 18 $BiCuCD_6$ compounds with organic C cations (C = Cs_2/MA_2/FA_2/CsMA/CsFA/MAFA, D = Cl/Br/I), and proposed $BiCuFA_2I_6$ as most promising solar absorber based on its optical and electronic properties. Ding et al. [81] systematically screened 760 double perovskites of chemical formula $A^{2+}B^{2+}Cs_2D_6$. Structural stability was evaluated based on the Goldschmidt tolerance factor, the octahedral factor and *ab initio* Molecular Dynamics (AIMD). The band gap of the stable materials was calculated first with the PBE functional, and then with HSE06 for those compounds with a PBE gap in the range 0.5 eV to 1.75 eV. 14 direct gap compounds were found, 8 of them lead-free. Among these, $MnPtCs_2F_6$ showed the best optoelectronic and transport properties. Cai et al. [82] calculated the tolerance factors of 2000 double perovskites and screened the 1000 strutures predicted to be stable via PBE and HSE06 DFT calculations. 11 compounds were found to be dynamically and thermodynamically stable, and with a maximum theoretical power conversion efficiency higher than 10%. Bartel et al. [83] used a statistically learnt tolerance factor [48] to study the stability of 903 $ABCs_2Cl_6$ double perovskites, and calculated the thermodynamical stability of the resulting compounds. Of the 261 stable structures, 47 have a direct or quasi-direct band gap between 1 eV and 3 eV, and do not contain toxic elements. 26 of the 47 structures contain an alkali metal and a transition metal at the A/B position, and are referred to as triple-alkali perovskites. The electronic structure was further investigated via many-body GW_0

and Bethe-Salpeter Equation (BSE) calculations. Some of these compounds showed large exciton binding energies and electronic properties similar to those of layered materials or insulating bulk materials, and properties potentially tunable by sublattice mixing.

2 Electronic Structure Calculations

2.1 Density Functional Theory

The central quantity needed to quantum mechanically describe a system of atoms is the wavefunction [84–86]. A system at any moment is said to be in a state that can be described by a vector $|\Psi\rangle$ that contains all the information about said system. The wavefunction is the projection of the system's state in the spin-spatial coordinates space. For example, a system of electrons in a state $|\psi\rangle$ can be described by a wavefunction $\psi(\mathbf{x})$:

$$\langle \mathbf{x}|\psi\rangle = \psi(\mathbf{x}) \quad , \qquad \mathbf{x} = (\mathbf{r}, \omega) \quad . \tag{2.1}$$

Here we use $\mathbf{x}$ to indicate spin-spatial coordinates, and $\mathbf{r}$ and ω to indicate the spatial part and the spin part separately. The overall electronic wavefunction for a N-electron system can be written as a combination of N one-electron spin orbitals φ, that respects the Pauli exclusion principle and the principle of antisymmetry of fermionic wavefunctions. This can be achieved by combining the spin-orbitals in a Slater determinant

$$\langle \mathbf{x}|\psi\rangle = \psi(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N) = \frac{1}{\sqrt{N!}} \begin{vmatrix} \varphi_1(\mathbf{x}_1) & \varphi_2(\mathbf{x}_1) & \dots & \varphi_N(\mathbf{x}_1) \\ \varphi_1(\mathbf{x}_2) & \varphi_2(\mathbf{x}_2) & \dots & \varphi_N(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_1(\mathbf{x}_N) & \varphi_2(\mathbf{x}_N) & \dots & \varphi_N(\mathbf{x}_N) \end{vmatrix} \quad . \tag{2.2}$$

Each spin orbital has a spatial part $\chi(\mathbf{r})$ and a spin part $\sigma(\omega) = \alpha(\omega)$ or $\beta(\omega)$ (i.e. spin up or spin down)

$$\varphi_l(\mathbf{x}_k) = \chi_l(\mathbf{r}_k)\sigma_l(\omega_k) \quad . \tag{2.3}$$

Here the index l identifies the spin orbital and the index k identifies the electron. Spin orbitals respect the orthonormalithy condition

$$\int d\mathbf{x}_i \varphi_k^*(\mathbf{x}_i)\varphi_l^*(\mathbf{x}_i) = \int d\omega_i \sigma_k(\omega_i)\sigma_l(\omega_i) \int d\mathbf{r}_i \chi_k^*(\mathbf{r}_i)\chi_l(\mathbf{r}_i) = \delta_{\sigma_k,\sigma_l}\delta_{\chi_k,\chi_l} \quad , \tag{2.4}$$

where

$$\delta_{i,j} = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases} \quad . \tag{2.5}$$

The spatial part can be expanded in as a linear combination of (normalized) atomic orbitals (LCAO)

$$\chi_i(\mathbf{r}) = \sum_{\mu=1}^{K} (\mathbf{C})_{\mu i}\phi_\mu(\mathbf{r}) \quad . \tag{2.6}$$

Every spin orbital can be written as a combination of a fixed set of atomic orbitals ϕ (i.e. the basis set) with the appropriate coefficient. The coefficients can be found by solving the Schrödinger's equation that describes the system.

If we consider a system composed of N electrons and M nuclei, we can describe their interaction with the following Hamiltonian operator

$$\hat{\mathcal{H}} = \overbrace{-\sum_{i=1}^{N}\frac{1}{2}\nabla_i^2}^{\hat{\mathcal{T}}_e} \overbrace{-\sum_{A=1}^{M}\frac{1}{2M_A}\nabla_A^2}^{\hat{\mathcal{T}}_N} \overbrace{-\sum_{i=1}^{N}\sum_{A=1}^{M}\frac{Z_A}{r_{iA}}}^{\hat{\mathcal{V}}_{Ne}} \overbrace{+\frac{1}{2}\sum_{\substack{i,j=1\\ i\neq j}}^{N}\frac{1}{r_{ij}}}^{\hat{\mathcal{V}}_{ee}} \overbrace{+\frac{1}{2}\sum_{\substack{A,B=1\\ A\neq B}}^{M}\frac{Z_A Z_B}{R_{AB}}}^{\hat{\mathcal{V}}_{NN}} \tag{2.7}$$

where the five terms indicate, respectively, the kinetic energy of the electrons and nuclei, the electron-nucleus attraction, the electron-electron repulsion and the nucleus-nucleus repulsion (here in atomic units). With this formulation of the Schrödinger's equation, the electronic and nuclear degrees of freedom are coupled to each other and can not be separated. However, it is common to apply the Born-Oppenheimer approximation and express the total wavefunction as a product of two independent wavefunctions (one for electrons and one for the nuclei) calculated at a fixed nuclear configuration $\mathbf{R}_A$

$$\Psi(\{\mathbf{x}_i\},\{\mathbf{R}_A\}) = \psi_e(\{\mathbf{x}_i\};\{\mathbf{R}_A\}) \times \psi_N(\{\mathbf{R}_A\}) \quad . \tag{2.8}$$

The two wavefunctions ψ_e and ψ_N are calculated as eigenfunctions of the electronic and nuclear Hamiltonians $\hat{\mathcal{H}}_e$ and $\hat{\mathcal{H}}_N$

$$\begin{aligned} \hat{\mathcal{H}}_e\psi_e(\{\mathbf{x}_i\};\{\mathbf{R}_A\}) = (\hat{\mathcal{T}}_e + \hat{\mathcal{V}}_{Ne} + \hat{\mathcal{V}}_{ee})\psi_e(\{\mathbf{x}_i\};\{\mathbf{R}_A\}) = \mathcal{E}_e\psi_e(\{\mathbf{x}_i\};\{\mathbf{R}_A\}) \\ \hat{\mathcal{H}}_N\psi_N(\{\mathbf{R}_A\}) = (\hat{\mathcal{T}}_N + \hat{\mathcal{V}}_{NN})\psi_N(\{\mathbf{R}_A\}) = \mathcal{E}_N\psi_N(\{\mathbf{R}_A\}) \quad . \end{aligned} \tag{2.9}$$

The coupling term

$$C = -\frac{1}{2}\sum_{A=1}^{M}\frac{1}{M_A}(\nabla_A^2\psi_e)\psi_N + (\nabla_A\psi_e)\nabla_A\psi_N \tag{2.10}$$

is therefore neglected. If we apply $\hat{\mathcal{H}}$ to $\psi_e\psi_N$ we notice that $\psi_e\psi_N$ is not an eigenfunction of $\hat{\mathcal{H}}$

$$\hat{\mathcal{H}}\psi_e\psi_N = C + \mathcal{E}\psi_e\psi_N \tag{2.11}$$

but considering that the motion of the nuclei is orders of magnitude slower than that of the electrons, it is reasonable in many cases to consider them as static respect to the electrons, and not include C in the further calculations.

Additionally, the N-electrons wavefunction depends on 4N coordinates (3 spatial and one spin degrees of freedom per electron) and this is often untreatable except in the case of very small systems. One possible approach to mitigate this problem is given by Density Functional Theory (DFT). Within the DFT framework, the Schrödinger's equation is reformulated in terms of electron density, which depends only on 3 spatial coordinates plus the spin. In this context, the density operator is defined as

$$\hat{n}(\mathbf{r}) = \sum_{i=1}^{N} \delta(\mathbf{r} - \mathbf{r}_i) \Rightarrow n(\mathbf{r}) = \langle \psi | \hat{n}(\mathbf{r}) | \psi \rangle \tag{2.12}$$

and it allows to rewrite the electron-nucleus attraction operator $\hat{\mathcal{V}}_{Ne}$ as

$$\begin{aligned} \hat{\mathcal{V}}_{Ne} &\rightarrow \sum_{i=1}^{N} v(\mathbf{r}_i) \\ \sum_{i=1}^{N} v(\mathbf{r}_i) &= \sum_{i=1}^{N} \int \mathbf{dr} v(\mathbf{r}) \delta(\mathbf{r}_i - \mathbf{r}) = \int \mathbf{dr} v(\mathbf{r}) \hat{n}(\mathbf{r}). \end{aligned} \tag{2.13}$$

In DFT this term is known as external potential.

The foundations of the DFT framework are two theorems by Hohenberg and Kohn (HK) [87], which affirm that:

- The ground state wavefunction of an N-electron system is a unique functional of the ground state electron density, which in turn determines (up to a constant shift) the local potential experienced by the system
- The ground state energy of an N-electron system is a unique functional $\mathcal{E}_0[n_0]$ of the ground state electron density $n_0(\mathbf{r})$, and this electron density is the one that minimizes the energy functional $\mathcal{E}[n]$.

For an electronic system subject to the external potential $v(\mathbf{r})$ with electron density $n(\mathbf{r})$ the energy functional is

$$\begin{aligned} \mathcal{E}_v[n] &= \langle \psi[n] | \hat{\mathcal{T}}_e + \hat{\mathcal{V}}_{ee} + \hat{\mathcal{V}}_{Ne} | \psi[n] \rangle \\ &= \langle \psi[n] | \hat{\mathcal{T}}_e + \hat{\mathcal{V}}_{ee} | \psi[n] \rangle + \int \mathbf{dr} v(\mathbf{r}) n(\mathbf{r}) \\ &= \mathcal{F}[n] + \int \mathbf{dr} v(\mathbf{r}) n(\mathbf{r}) \end{aligned} \tag{2.14}$$

$\mathcal{F}[n]$ is known as internal energy functional because it's not explicitly dependent on the specific geometry of the system.

Kohn and Sham (KS) [88] re-framed DFT to make it tractable in practice. The HK theorems allow us to construct a fictitious system of non-interacting electrons ($\hat{\mathcal{V}}_{ee} = \hat{0}$) in a potential $v_{KS}(\mathbf{r})$ chosen such that its ground state electron density equals that of the real fully interacting system. The Hamiltonian for this system can therefore be expressed as

$$\hat{\mathcal{H}}_{KS} = \hat{\mathcal{T}}_e + \hat{\mathcal{V}}_{KS} \quad . \tag{2.15}$$

For a system of non-interacting electrons, the kinetic energy can be defined exactly, while the internal energy functional can be expressed as

$$\begin{aligned} \mathcal{F}[n] &= \mathcal{T}_{KS}[n] + (\mathcal{F}[n] - \mathcal{T}_{KS}[n]) \\ &= \langle \psi_{KS}[n] | \hat{\mathcal{T}}_e | \psi_{KS}[n] \rangle + \mathcal{E}_{Hxc}[n] \end{aligned} \tag{2.16}$$

The term $\mathcal{E}_{Hxc}[n]$ includes the Hartree energy (i.e. classical electrostatic repulsion for a certain charge distribution) and the exchange-correlation energy.

$$\mathcal{E}_{Hxc}[n] = \mathcal{E}_H[n] + \mathcal{E}_{xc}[n] = \frac{1}{2} \int \int \mathrm{d}\mathbf{r}_1 \mathrm{d}\mathbf{r}_2 \frac{n(\mathbf{r}_1) n(\mathbf{r}_2)}{r_{12}} + \mathcal{E}_{xc}[n] \quad . \tag{2.17}$$

The exchange energy represents the repulsion between electrons with the same spin as a consequence of the Pauli exclusion principle. The correlation reflects the fact that the motion of one electron is affected by all the other electrons in the system

The final expression of the ground state energy is therefore

$$\mathcal{E}_0 = \mathcal{T}_{KS}[n_0] + \mathcal{E}_{Hxc}[n_0] + \int \mathrm{d}\mathbf{r} v(\mathbf{r}) n_0(\mathbf{r}). \tag{2.18}$$

The exchange-correlation term appears only in a quantum mechanical treatment of the electrons, and needs to be approximated. The following are well known classes of exchange-correlation functionals:

- Local Density Approximation (LDA): the exchange-correlation energy density in one point in space depends only on the value of the electron density in that specific point. Its expression equals that of a uniform electron gas having the same electron density.
- Generalized Gradient Approximation (GGA): the exchange-correlation energy depends both on the local value of the density as in LDA, but also on its gradient.

$$\epsilon_{x/c}^{GGA}[n(\mathbf{r})] = \epsilon_{x/c}^{LDA}[n(\mathbf{r})] + \Delta\epsilon_{x/c}^{GGA}[n(\mathbf{r})] \left[\frac{\nabla n(\mathbf{r})}{n(\mathbf{r})^{4/3}} \right] \tag{2.19}$$

- meta-GGA: the exchange-correlation energy depends on the kinetic-energy density.

$$\tau(\mathbf{r}) = \sum_{i=1}^{N} \frac{1}{2} |\nabla \psi_{KS}(\mathbf{r})|^2 \tag{2.20}$$

- Hybrid functionals: the exchange-correlation energy is a linear combination of exact exchange, that can be calculated following the Hartree-Fock method on Kohn-Sham states, and a DFT exchange-correlation energy calculated from some other functional.

$$E_{xc} = (1-\alpha)E_{xc}^{DFT} + \alpha E_x^{HF} \tag{2.21}$$

  The coefficient α has to be determined and varies across different functionals (e.g. PBE0 has $\alpha = 0.25$)

- Range separated hybrid functionals: only short range (SR) exchange is added, while at the long range the exchange is fully GGA

$$E_{xc} = (1-\alpha)E_x^{DFT,SR}(\mu) + \alpha E_x^{HF,SR}(\mu) + E_x^{DFT,LR}(\mu) + E_c^{DFT} \tag{2.22}$$

  Here μ indicates the cutoff between short range (SR) and long range (LR).

The choice of the functional is critical, and and needs to be made with great care. Despite the involved approximations, there are nowadays many functionals available that have been used countless times on different chemical systems, and lead to accurate results in many fields of computational chemistry.

After choosing the appropriate exchange-correlation functional and basis set, the ground state energy and density can be found by with an iterative scheme known as self-consistent field, that involves the following steps:

1) Make an initial guess for the coefficients of the basis set expansion
2) Construct the first density
3) Use the density to construct the KS Hamiltonian
4) Solve to find new coefficients and the ground state energy
5) Use the new coefficients to construct an updated density
6) Repeat steps 3-5 until the difference between the current density and energy and those at the previous step are below a user-defined convergence threshold

2.2 Electrons in Periodic Systems

Crystalline materials are composed of atoms (or molecules) that repeat regularly in a periodic structure [89]. Mathematically, this can be described with a lattice $\mathcal{L}$ and a basis $\mathcal{B}$. The lattice is a set of points in space that forms an infinite periodic pattern

$$\mathcal{L}(x) = \sum_n \delta(x - n\mathbf{a}) \quad , \tag{2.23}$$

from integer combinations of linearly independent basis vectors $\mathbf{a}$. Every pair of points in the lattice are connected by vectors of the type

$$\mathbf{R} = n_1\mathbf{a}_1 + n_2\mathbf{a}_2 + n_3\mathbf{a}_3 \tag{2.24}$$

The basis is a motif, such as an atom, a group of atoms, or a molecule, that is repeated in correspondence to each point of the lattice. The convolution of lattice and basis forms the crystal $\mathcal{C}$ (Figure 2.1)

$$\mathcal{C}(x) = \mathcal{L}(x) \star \mathcal{B}(x) = \int_{-\infty}^{\infty} \mathcal{L}(x')\mathcal{B}(x - x')dx' = \sum_n \mathcal{B}(x - n\mathbf{a}) \quad . \tag{2.25}$$

Three lattice vectors (in a 3-dimensional lattice) define a unit cell that can be trans-

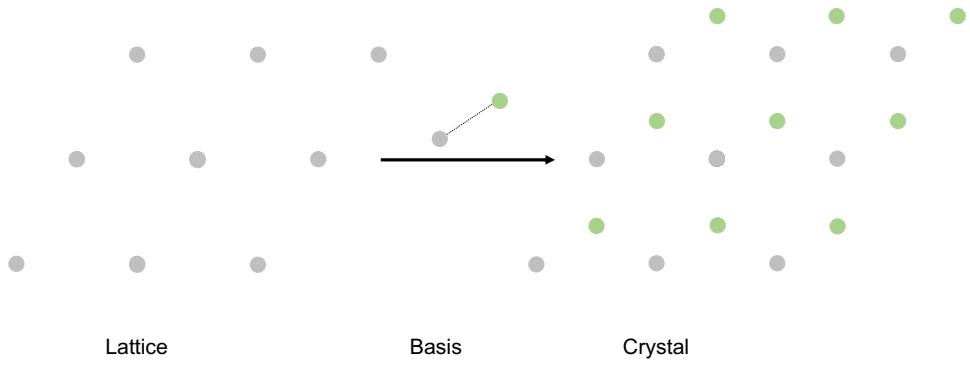


Figure 2.1 Crystal with a honeycomb structure formed from the convolution of a hexagonal lattice with a biatomic basis

lated to fill up the entire space without any overlap with other cells, and without leaving any void. The smallest possible unit cell, i.e. the one that contains only one lattice point, is known as primitive cell. It is possible to define different primitive cells on the same lattice, and such cells don't necessarily reflect the symmetry of the underlying lattice. A specific type of primitive cell that reflects the symmetry of the lattice is the Wigner-Seitz cell. This kind of cell is defined by choosing one lattice point as center and placing planes that bisect the distance between this point and its nearest neighbors. The volume inside these planes contains all the points in space that are closer to the central lattice point than to any other lattice point (Figure 2.2).

The filling of all space with a periodic arrangement of cells is possible only for the so-called Bravais lattices, i.e. lattices that have only translations and two-, three-, four-

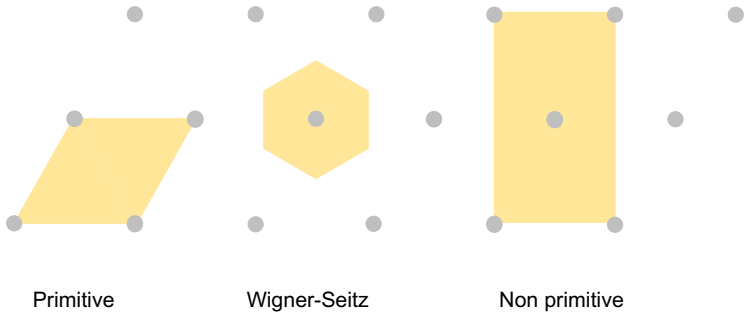


Figure 2.2 Examples of primitive, Wigner-Seitz and non primitive cells in a 2-dimensional hexagonal lattice

, and six-fold rotations as symmetry operations. For each Bravais lattice, a reciprocal lattice can be defined. The basis vectors of the reciprocal lattice are

$$\mathbf{b}_i = 2\pi \frac{\mathbf{a}_j \times \mathbf{a}_k}{\mathbf{a}_i \cdot (\mathbf{a}_j \times \mathbf{a}_k)} = \frac{2\pi}{V_{cell}} \mathbf{a}_j \times \mathbf{a}_k \tag{2.26}$$

and satisfy the condition

$$\mathbf{a}_i \cdot \mathbf{b}_j = 2\pi\delta_{ij} \quad . \tag{2.27}$$

Mathematically, the reciprocal lattice is the Fourier transform of the real space lattice (or direct lattice) and it can be interpreted as the lattice as seen in momentum space. It is often convenient to work in reciprocal space, as many physical entities in periodic crystals are derived from planewaves that have the periodicity of the lattice. This is possible only for a set of wavevectors $\mathbf{G}$ that respect the condition

$$e^{i\mathbf{G}(\mathbf{R}+\mathbf{r})} = e^{i\mathbf{G}\mathbf{r}} \Rightarrow \mathbf{G} \cdot \mathbf{R} = 2\pi n, \quad n \in \mathbb{Z} \tag{2.28}$$

From this we can see that $\mathbf{G}$ vectors are actually integer linear combinations of reciprocal lattice vectors

$$\mathbf{G} = n_1\mathbf{b}_1 + n_2\mathbf{b}_2 + n_3\mathbf{b}_3 \quad , \tag{2.29}$$

and that the reciprocal lattice is also a Bravais lattice. The Wigner-Seitz cell of the reciprocal lattice is also known as first Brilluoin zone.

The concept of lattice is useful to describe the behavior of valence electrons in the periodic potential formed by the ions in a crystal. According to Bloch's theorem [90,

91], when the wavefunction is subject to a translation $\mathbf{T}$ by a lattice vector $\mathbf{R}$, then the resulting wavefunction can only change by a phase factor

$$\mathbf{T}_{\mathbf{R}}\psi(\mathbf{r}) = \psi(\mathbf{r}+\mathbf{R}) = e^{i\mathbf{k}\mathbf{R}}\psi(\mathbf{r}) \quad . \tag{2.30}$$

This holds true if the wavefunction has the form

$$\psi_{\mathbf{k}}(\mathbf{r}) = e^{i\mathbf{k}\mathbf{r}}u_{\mathbf{k}}(\mathbf{r}) \quad , \quad \text{with} \quad u_{\mathbf{k}}(\mathbf{r}) = u_{\mathbf{k}}(\mathbf{r}+\mathbf{R}) \quad , \tag{2.31}$$

i.e. when the wavefunction is a planewave modulated by a periodic function $u_{\mathbf{k}}$. These are known as Bloch functions. Given that $e^{i\mathbf{G}\mathbf{R}} = 1$, the wavevector $\mathbf{k}$ can always be considered as confined inside the first Brillouin zone, because a wavefunction $\psi_{\mathbf{k}'}$ with wavevector $\mathbf{k}' = \mathbf{k} + \mathbf{G}$ larger than the first Brillouin zone, would be equivalent to $\psi_{\mathbf{k}}$.

Until now we have relayed on the concept of an infinite lattice, although a real crystal is finite. We can still consider the crystal as "infinite" if we impose periodic boundary conditions on the wavefunction, i.e. if the wavefunction translated for a large number of cells $N = N_1N_2N_3$ (as large as in a macroscopic crystal) equals the wavefunction in the first cell

$$\psi_{\mathbf{k}}(\mathbf{r} + (N_1\mathbf{a}_1 + N_2\mathbf{a}_2 + N_3\mathbf{a}_3)) = e^{i\mathbf{k}(N_1\mathbf{a}_1+N_2\mathbf{a}_2+N_3\mathbf{a}_3)}\psi_{\mathbf{k}}(\mathbf{r}) = \psi_{\mathbf{k}}(\mathbf{r}) \quad . \tag{2.32}$$

This is true if

$$e^{i\mathbf{k}(N_1\mathbf{a}_1+N_2\mathbf{a}_2+N_3\mathbf{a}_3)}\psi_{\mathbf{k}}(\mathbf{r}) = 1 \Rightarrow \mathbf{k} = \frac{n_1}{N_1}\mathbf{b}_1 + \frac{n_2}{N_2}\mathbf{b}_2 + \frac{n_3}{N_3}\mathbf{b}_3 \quad . \tag{2.33}$$

Equation 2.33 suggests that $\mathbf{k}$ needs to be quantized, however in a macroscopic crystal the number of cells will be so large that $\mathbf{k}$ can be considered continuous.

Bloch functions are one-electron wavefunctions and solutions of a Schrödinger equation with a kinetic energy term and a periodic potential

$$\left[-\frac{1}{2}\nabla^2 + V(\mathbf{r})\right]\psi = \mathcal{E}\psi \quad , \quad \text{with} \quad V(\mathbf{r}) = V(\mathbf{r}+\mathbf{R}) \quad . \tag{2.34}$$

When restricted to the first Brillouin zone, this gives rise to a family of eigenfunctions $\psi_{n\mathbf{k}}$ with eigenvalues $\mathcal{E}_{n\mathbf{k}}$. For each n the energies in the N points $\mathbf{k}$ define an energy band. Each band in the first Brillouin zone can accomodate $2N$ electrons (N with spin up and N with spin down). Also to calculate the total energy of the crystal with contributions from all electrons, it is sufficient to let $\mathbf{k}$ vary inside the first Brillouin zone.

To solve the Schrödinger equation in a periodic potential [92, Chapter 4] we can expand the potential in a Fourier series using the reciprocal space vectors $\mathbf{G}$ to respect the periodicity of the lattice

$$V(\mathbf{r}) = \sum_{\mathbf{G}} V_{\mathbf{G}} e^{i\mathbf{G}\mathbf{r}} \quad . \tag{2.35}$$

The same can be done with the periodic part of the Bloch wavefunctions (here we temporarily drop the labels $n\mathbf{k}$ for ease of notation)

$$u(\mathbf{r}) = \sum_{\mathbf{G}} C_{\mathbf{G}} e^{i\mathbf{G}\mathbf{r}} \quad . \tag{2.36}$$

This leads to the following equation

$$\left[\mathcal{E} - V_0 - \frac{\hbar^2}{2m_e}(\mathbf{k}+\mathbf{G})^2\right] C_{\mathbf{G}} = \sum_{|\mathbf{G}'|\neq 0} V_{\mathbf{G}'} C_{\mathbf{G}-\mathbf{G}'} \quad . \tag{2.37}$$

Here we explicitly write the reduced Planck constant $\hbar$ and the electron mass m_e instead of using atomic units. If $V_{\mathbf{G}} = 0$ for all $|\mathbf{G}| \neq 0$, we obtain a free electron model, with energy

$$\mathcal{E} = V_0 + \frac{\hbar^2}{2m_e}(\mathbf{k}+\mathbf{G})^2 \quad . \tag{2.38}$$

In this case the dispersion relation $\mathcal{E}(\mathbf{k})$ has a parabolic shape with a kinetic energy $\epsilon_{\mathbf{k}+\mathbf{G}} = \frac{\hbar^2}{2m_e}(\mathbf{k}+\mathbf{G})^2$ and an offset V_0. This is also the zeroth order approximation in perturbation theory. If $V_{\mathbf{G}}$ is small for all $|\mathbf{G}| \neq 0$, we can use the first order approximation to get an additional term

$$\mathcal{E} = V_0 + \epsilon_{\mathbf{k}+\mathbf{G}} - \sum_{|\mathbf{G}|\neq 0} \frac{|V_{\mathbf{G}}|^2}{\epsilon_{\mathbf{k}+\mathbf{G}} - \epsilon_{\mathbf{k}}} \quad . \tag{2.39}$$

This is the nearly free electron model, i.e. for an electron that moves in a weak periodic potential. This holds true as long as $|\mathbf{k}+\mathbf{G}| \neq |\mathbf{k}|$. For the case where $|\mathbf{k}+\mathbf{G}| \simeq |\mathbf{k}|$, the energy becomes

$$\mathcal{E} = V_0 + \frac{\epsilon_{\mathbf{k}} + \epsilon_{\mathbf{k}+\mathbf{G}}}{2} \pm \sqrt{|V_{\mathbf{G}}|^2 + \left(\frac{\epsilon_{\mathbf{k}} - \epsilon_{\mathbf{k}+\mathbf{G}}}{2}\right)^2} \tag{2.40}$$

and for the specific case where $|\mathbf{k}+\mathbf{G}| = |\mathbf{k}|$, the energy is

$$\mathcal{E} = V_0 + \epsilon_{\mathbf{k}} \pm |V_{\mathbf{G}}| \tag{2.41}$$

If we introduce $\mathbf{q} = \frac{\mathbf{G}}{2} + \mathbf{k}$, and expand around small values of $\mathbf{q}$ we get

$$\mathcal{E} = V_0 + \epsilon_{\frac{\mathbf{G}}{2}} + \epsilon_{\mathbf{q}} \pm |V_{\mathbf{G}}| \left(1 + \frac{\epsilon_{\mathbf{G}}}{2|V_{\mathbf{G}}|^2}\epsilon_{\mathbf{q}}\right) \tag{2.42}$$

We can set the zero at $\mathcal{E}_0 = V_0 + \epsilon_{\frac{\mathbf{G}}{2}} - |V_{\mathbf{G}}|$ and define

$$\frac{1}{m^*_{\pm}} = \frac{1}{m_e}\left(1 \mp \frac{\epsilon_{\mathbf{G}}}{2|V_{\mathbf{G}}|}\right) \quad , \tag{2.43}$$

and finally write the two values of the energy as

$$\begin{aligned}\mathcal{E}_- &= \frac{\hbar^2 q^2}{2m_-^*} = -\frac{\hbar^2 q^2}{2m_v} \\ \mathcal{E}_+ &= E_{gap} + \frac{\hbar^2 q^2}{2m_+^*} = E_{gap} + \frac{\hbar^2 q^2}{2m_c} \quad ,\end{aligned} \tag{2.44}$$

where $E_{gap} = 2|V_{\mathbf{G}}|$ is known as band gap. The mass $m_\pm^*$ is known as effective mass. Using this concept, we can approximate the motion of an electron in a weak periodic potential as that of a free electron (i.e. its energy is kinetic and has a parabolic dispersion) but with a modified mass that takes into account the interactions with the lattice. The lowest energy band above E_{gap} is known as conduction band and has a positive curvature and therefore a positive effective mass $m_+^* = m_c$. The highest band below the band gap is the valence band, and has negative curvature and therefore negative effective mass $m_-^* = -m_v$.

It is also possible that in some directions the bands overlap, so the material overall doesn't have a gap. Based on this we can classify the materials in

- *Metals*:
  - If the material has an odd number of valence electrons per unit cell, and remembering that a band can accomodate $2N$ electrons, then the band will be only half filled, and even a small amount of energy can promote the electrons to the higher continuous energy states. Some examples of these metals are Li, Na, K, Rb, Cs, Cu, Ag, and Au.
  - If the material has an even number of valence electrons but the bands overlap in some direction, there will be empty states in the lower band and filled states in the upper band. If this number is of the order of N, the material will also be metallic. Examples of these case are Zn, Cd, Ca, Mg, and Ba.
- *Semimetals*: If the bands have a small overlap and the unit cell contains an even number of valence electrons, the number of empy states in the lower band and filled states in the upper band will be much smaller than N, so the materials is called semimetal. Examples of semimetals are As, Sb, Bi, Sn, and graphite.
- *Semiconductors*: If there is a gap that allows a few electrons to occupy the conduction band at room temperature (approximately between 0.1 eV and 2eV) the materials is a semiconductor. Examples of semiconductors are Ge, Si, InSb, GaAs, AlSb, InAs, InP.
- *Insulators*: If the gap is large so the electrons can't be excited to the conduction band the material is an insulator. Examples of insulators are diamond and Al_2O_3

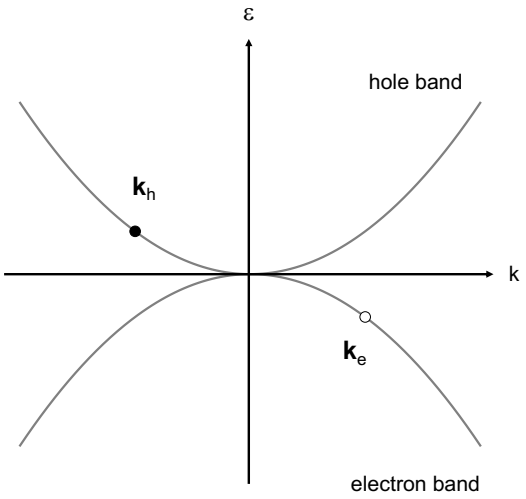


Figure 2.3 Electron and hole bands. Adapted from Figure 8 Chapter 8 in reference [90]

2.3 Electronic Transport in Semiconductors

The dispersion relation $\epsilon_e(\mathbf{k})$ indicates the effect of the crystal potential on the motion of electrons. These electrons will move with velocity

$$\mathbf{v} = \frac{\nabla_{\mathbf{k}}\epsilon(\mathbf{k})}{\hbar} \quad . \tag{2.45}$$

When an external force is applied, such as an electromagnetic field, this will be equal to

$$\mathbf{F} = \hbar\frac{d\mathbf{k}}{dt} \quad . \tag{2.46}$$

When all the states in a band are filled, the electrons can't carry a current even if an electromagnetic field is applied (unless this field has enough energy to promote electrons to the next unoccupied band and leave the current band partially filled). Because of the inversion symmetry $\mathbf{k} \rightarrow -\mathbf{k}$ in the Brillouin zone, the sum of the wavevectors of all the electrons in a filled band is zero. If one electron is missing, i.e. if one electron has been excited to the upper band via temperature effects or absorption of light, a current can be carried, and the total wavevector of the system becomes $-\mathbf{k}_e$. In this case, instead of treating the collective motion of all the remaining electrons, it's possible to write the equations of motion of a single particle know as *hole* that correspond to a missing electron. The wavevector of the hole is $\mathbf{k}_h = -\mathbf{k}_e$ and its dispersion relation $\epsilon_h(\mathbf{k}_h) = -\epsilon_e(\mathbf{k}_e)$ (Figure 2.3). Given that $\nabla_{\mathbf{k}}\epsilon(\mathbf{k}_h) = \nabla_{\mathbf{k}}\epsilon(\mathbf{k}_e)$, the velocity of the hole will be

$$\mathbf{v}_h = \mathbf{v}_e \tag{2.47}$$

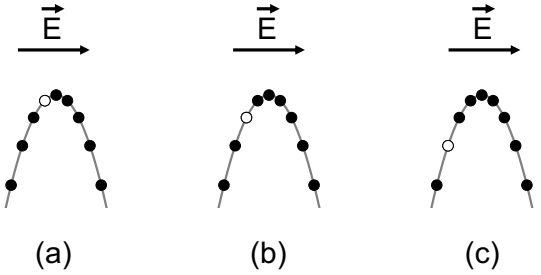


Figure 2.4 Movement of electrons and a hole in an electric field. Adapted from Figure 9 Chapter 8 in reference [90]

The hole moves in the same direction with the same velocity of the electron. The equation of motion of an electron and a hole in an electromagnetic field are

$$\begin{aligned} \hbar\frac{d\mathbf{k}_e}{dt} &= -e\left(\mathbf{E}+\mathbf{v}_e\times\mathbf{B}\right) \\ \hbar\frac{d\mathbf{k}_h}{dt} &= e\left(\mathbf{E}+\mathbf{v}_h\times\mathbf{B}\right) \quad . \end{aligned} \tag{2.48}$$

The equation of motion of the hole is that of a positively charged particle of charge e. When the field is applied and the bands are isotropic the acceleration of the particles is

$$\frac{dv}{dt}=\frac{1}{\hbar}\left(\frac{d^2\epsilon}{dk^2}\frac{dk}{dt}\right)=\left(\frac{1}{\hbar^2}\frac{d^2\epsilon}{dk^2}\right)F=\frac{1}{m^*}F \quad , \tag{2.49}$$

or for an anisotropic case

$$\frac{dv_i}{dt}=\left(\frac{1}{\hbar^2}\frac{d^2\epsilon}{dk_i dk_j}\right)F_j=\left(\frac{1}{m^*}\right)_{ij}F_j \tag{2.50}$$

which shows again how the electrons or holes move as free particles but with a mass that depends on the curvature of the band in which they reside.

3 Artificial Neural Networks

3.1 Feedforward Neural Networks

Artificial neural networks [93, 94] are used to approximate a non-linear function that maps an input **x** to a label **y**, which could either be a category or take continuous values. The parameters of the mapping need to be learned from available data, i.e. the training set. The basic building block of a neural network is the neuron, a unit that takes in an input **x**, applies first a linear transformation by multiplying it by a weight matrix **W** and adding a bias **b**, and successively applies a non-linear transformation σ to the output. In analogy to the action of biological neurons this function is known as the activation function. Figure 3.1 shows an example of a neuron i that takes three scalar inputs and returns the output $y_i = \sigma\left(\sum_{j=1}^{3} w_{ij}x_j + b_i\right)$. Several neurons stacked and connected to each other form the network architecture. The layers in between the input and output are called hidden layers:

$$\begin{aligned} \mathbf{h}^{(1)} &= \sigma_1(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}) \\ \mathbf{h}^{(2)} &= \sigma_2(\mathbf{W}^{(2)}\mathbf{h}^{(1)} + \mathbf{b}^{(2)}) \\ &\dots \\ \hat{\mathbf{y}} &= \sigma_N(\mathbf{W}^{(N)}\mathbf{h}^{(N-1)} + \mathbf{b}^{(N)}) \quad . \end{aligned} \tag{3.1}$$

Figure 3.2 represents a simple example of a neural network with two hidden layers. Each connection represents the action of multiplying by the corresponding weights, adding the bias and applying the non-linear transformation. When all the units of a layer are connected to the next, such a layer is referred to as fully connected. The activation function allows to learn more complex mappings respect to a simple linear regression. In fact, a network consisting of a sequence of linear layers without activations would be simply equivalent to a single linear transformation where the weight matrix is the product of the weight matrices in all the layers, and the bias is a linear combination of the biases in each layer. In principle, any non-linear function could be used as activation, however these are normally selected based on ease of calculation, the range of their output, and their gradient. In particular, the gradient of the activation functions affects the rate at which the network parameters will be updated during learning.

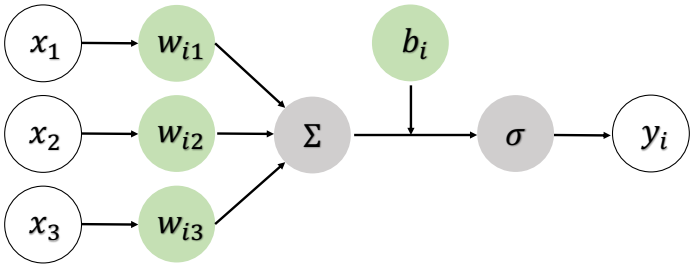


Figure 3.1 Single neuron in a neural network

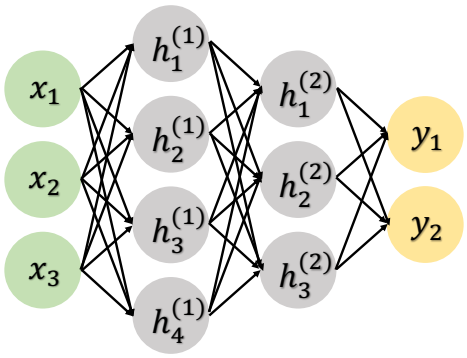


Figure 3.2 Feed forward neural network with 2 hidden layers

Some examples of non-linear activation functions (Figure 3.3) are the following:

- The rectified linear unit (ReLU) returns 0 if the input is negative and the input itself if positive. This activation function is the simplest and allows fast calculations, because its gradient is always either 0 or 1 depending on the sign of the input.

$$\text{ReLU}(x) = \max(0, x) \tag{3.2}$$

- Leaky ReLU is a variation of ReLU where the gradient below 0 takes a small positive value α

$$\text{LeakyReLU}(x) = \max(\alpha x, x) \tag{3.3}$$

- The exponential linear unit (ELU) is similar to LeakyReLU but it has an exponential form below zero.

$$\text{ELU}(x) = \begin{cases} x, \text{ for } x \geq 0 \\ \alpha(e^x - 1), \text{ for } x < 0 \end{cases} \tag{3.4}$$

- The sigmoid function returns values between 0 and 1, so it's often used when the output value is a probability.

$$\text{sigmoid}(x) = \frac{1}{1 + \exp(-x)} \tag{3.5}$$

- The hyperbolic tangent function returns values between -1 and 1, and it helps centering the data around 0, which can be beneficial for the learning process

$$\tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)} \tag{3.6}$$

- The softmax function returns positive values that sum up to 1, so it can be used in multi-class classification problems, where the sum of the probabilities for each class must be 1. Normally it's used only in the output layer.

$$\text{softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \tag{3.7}$$

The parameters of the network (weights and biases, here collectively noted as θ) are learned from the data by minimizing a loss function $J(\theta)$ that describes how well the output of the network matches the corresponding training label. Typical examples of loss functions are the mean absolute error or mean squared error for regression problems, and the cross entropy loss for classification problems. The parameters are first initialized as small random numbers, then the gradient of the loss function is

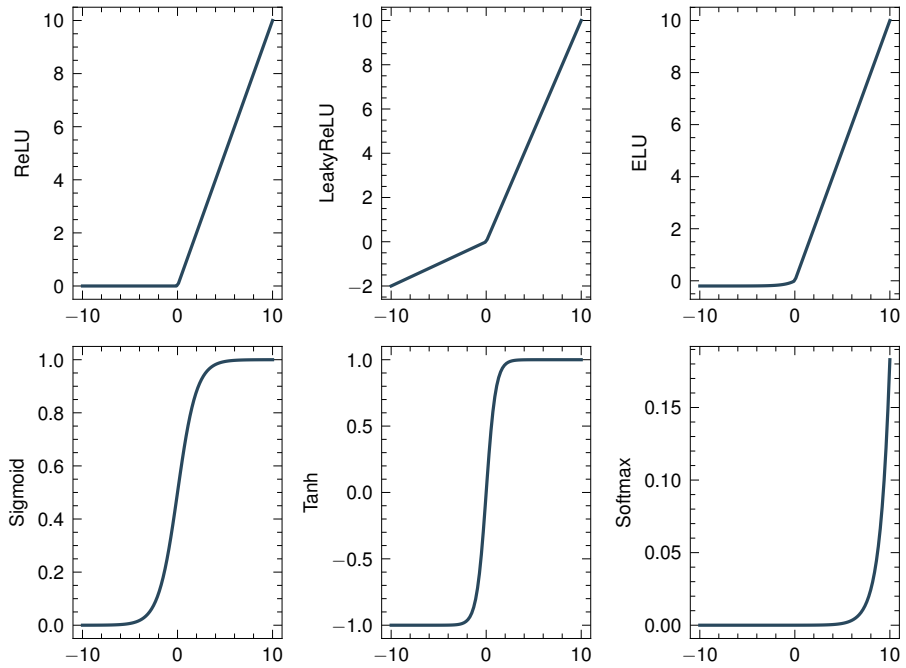


Figure 3.3 Some examples of activation functions. The input is a vector of 100 evenly spaced points in the interval [-10,10]

calculated with respect to said parameters, and the weights are iteratively updated to approach the minimum of the loss function:

$$\theta_t \leftarrow \theta_{t-1} - \eta \nabla_\theta J_t(\theta_{t-1}) \quad . \tag{3.8}$$

Here η indicates the learning rate and it controls the magnitude of the weights update, t is the update step, and ∇_θ is the gradient of the loss function respect to the parameters of the network. Given that the loss function is defined for a single datapoint, and the overall loss is the average of the loss function over the whole training set, one would need to calculate the gradient for each datapoint at each iteration. However, to reduce the computational cost, the gradient is approximated as the gradient calculated for a subset (or minibatch) randomly sampled at each iteration. This method is known as Minibatch Stochastic Gradient Descent (MSGD). The optimum size for the minibatches is a trade-off between accuracy in the estimation of the gradient, and frequency of the parameters update. The learning rate can be constant throughout the optimization or it can follow some different update schemes, for example it can start at higher values at the beginning of the optimization and then decrease while approaching the minimum. The gradient respect to the parameters in all the layers of the network is calculated via the chain rule starting from the output layer, as the output of each layer depends on the parameters of the previous layers. This procedure is known as backpropagation.

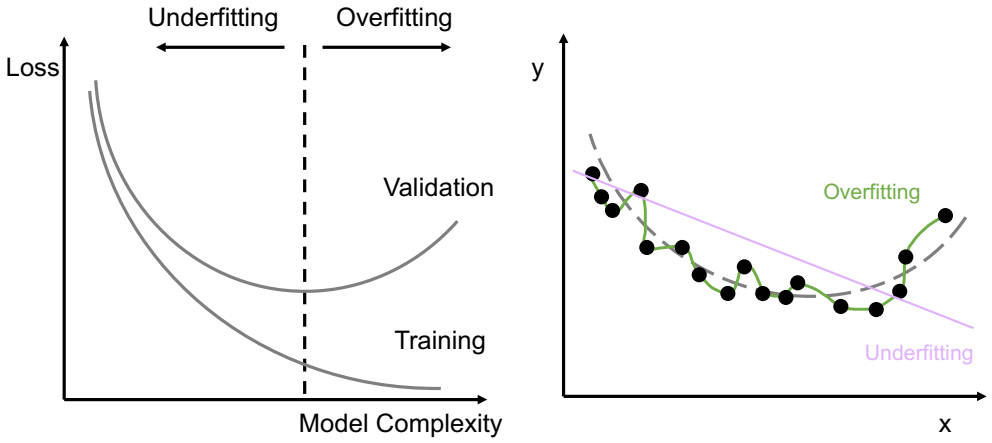


Figure 3.4 Loss function vs model complexity on a training and Validation set (left). Schematic example of an overfitting and underfitting model (right).

The objective of neural neural network-based models is not only to minimize the loss function on the training dataset, but to perform equally well on unseen data, i.e. it has to be able to generalize. A model that performs well only on the training set but not on an unseen validation set is said to be *overfitting*, as opposed to *underfitting*, i.e. a model that has high loss on both training and validation sets (Figure 3.4). The fitting capacity of the model, or its complexity, essentially depends on the number of parameters that it contains. Too low complexity will lead to underfitting the data, while too high complexity is likely to produce overfitting. When manipulating the number of parameters of the models doesn't solve the problem of overfitting, there are other *regularization* techniques that can be put in place.

- *Early stopping*: At the beginning of the training both the training and the validation loss should decrease. When the model starts memorizing the training set and fit the noise, the training loss will keep decreasing but the validation loss will either reach a plateau or might start to increase. By monitoring the two losses, or any other metric that represents the performance of the model, a number of iterations that gives the optimal performance can be found.

- *Weight decay*, consists in adding a penalty term to the loss function that is proportional to the square norm of the weights $\lambda||\mathbf{W}||_2^2$, with a hyperparameter λ to control the strength of the regularization effect. In this way, while minimizing the loss function the model will also keep the weights small, and have an

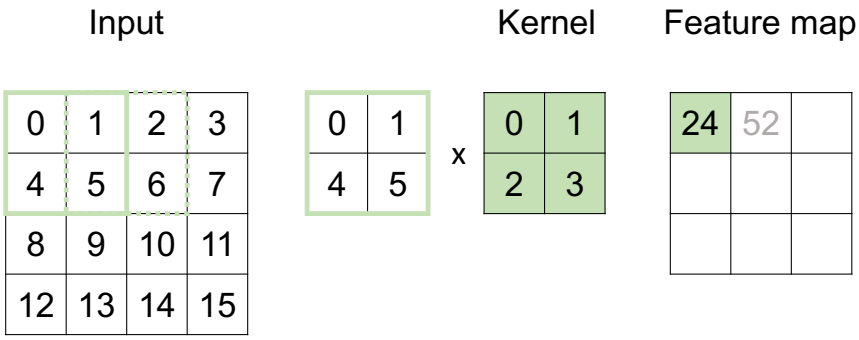


Figure 3.5 Example of convolution operation

overall low complexity. Weight decay can also be implemented with an L1 norm ($\lambda||\mathbf{W}||_1$), however, while the L2 norm drives all the weights to approach zero but with higher penalty for the largest ones and lower penalty for the smallest, the L1 norm pushes all the weights towards zero equally, so at the end of the training there will be some neurons with zero weight. This could also work as a feature selection method.

- *Dropout* consists in randomly replacing a certain fraction of the neurons in a hidden layer with zeros, with a certain probability p

$$h' = \begin{cases} 0, & \text{with probability } p \\ \dfrac{h}{1-p}, & \text{with probability } 1-p \end{cases} \tag{3.9}$$

In this way, the output of the layer that contains the dropout does not depend on the neurons that were dropped out, and the following neurons can't adapt to form a predefined path in the network. This a technique simulates the training of an ensamble of networks with a smaller number of neurons.

3.2 Convolutional Neural Networks

Convolutional Neural Networks (CNN) are neural networks designed to work on data that can be represented as a grid, for example an image represented as a 2-dimensional grid of pixels [93–95]. The task of a CNN is to learn the spatial structure of the input and extract features from it (for example edges and shapes of the objects in an image) to pass to the next layer. CNNs are built to respect the principles of

- *Locality*: the feature map around one pixel should be more sensitive to its local surroundings and less sensitive to the pixels far away from it.

- *Translational equivariance*: a shift in the original image should result in a corresponding shift in the feature map.

The hidden layers of a CNN can therefore be expressed as

$$\mathsf{H}_{i,j} = \sum_a \sum_b \mathsf{K}_{a,b}\mathsf{X}_{i+a,j+b} + \mathsf{b} \quad , \tag{3.10}$$

where **X** is the input, (i, j) is the position of each pixel, **b** is the bias and **K** is the convolutional kernel or filter. The elements of the kernel and the bias are parameters learned from the data. The kernel slides along the image and at each step the elements of the kernel and the corresponding elements of the image are multiplied and summed (Figure 3.5). The kernel can start from the corner of the image and remain within its margins, so that a kernel of size $k_h \times k_w$ on an image of size $x_h \times x_w$ can slide for $(x_h - k_h + 1) \times (x_w - k_w + 1)$ steps, or it's also possible to add a constant padding around the input to allow more steps. Additionally, the input can have more than 2 dimensions (such as 3 color channels in an image) and the hidden layers can have additional dimensions as well.

$$\mathsf{H}_{i,j,d} = \sum_a \sum_b \sum_c \mathsf{K}_{a,b,c,d}\mathsf{X}_{i+a,j+b,c} + \mathsf{b} \tag{3.11}$$

Convolution operations are followed by a non-linear activation function as in feedforward neural network, and can be also followed by a pooling operation. Pooling layers are used to select or aggregate information, and reduce the size of the feature map. They consist in a window of fixed size that slides over the feature map and computes one value using all the elements of the feature map included in the window. These values can be computed as the mean of the values in the window (average pooling), or the maximum among the values in the window (max pooling, Figure 3.6).

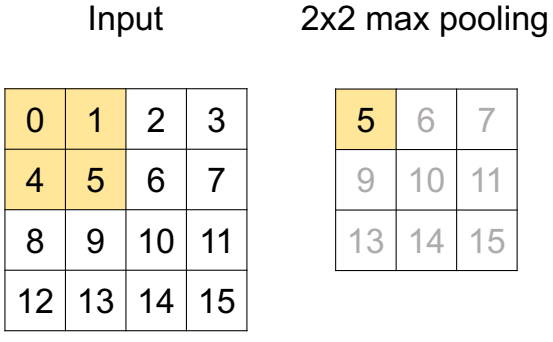


Figure 3.6 Example of max pooling

3.3 Message Passing Neural Networks

Message passing neural networks [96] are a type of neural network designed to work on graphs. A graph $G = (V, E)$ is a set of nodes V and edges E that connect pairs of nodes. A node v is associated with a feature vector x_v and the edge connecting nodes v and w with features e_{vw}. A graph can be directed if the order of the nodes in the pair matters, or undirected otherwise. For example, an undirected graph can be used as a representation of a molecule or crystal, where the nodes are atoms and the edges are bonds.

In a message passing neural network, the forward pass has two phases: a message passing phase that runs for T iterations and a readout phase. At each step t of the message passing phase, each node is associated to a hidden state h_v^t, with the initial hidden state h_v^0 being simply the input feature of the node v. A message function M_t is used to aggregate information from the hidden state of the node itself, from its nearest neighbors $N(v)$ and from the edges connecting the node to its neighbors. The sum of all the individual contributions m_v^{t+1} is known as message

$$m_v^{t+1} = \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_{vw}) \quad . \tag{3.12}$$

The hidden state is updated using an update function U_t to get the next hidden state

$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1}) \tag{3.13}$$

The readout phase then computes a feature vector for the whole graph, using the final hidden state of all the nodes

$$\hat{y} = R(h_v^T | v \in G) \tag{3.14}$$

Some examples of functions used for message, update and readout functions are the concatenation of feature vectors, sum, mean, learned matrices, activation functions, or neural networks. Figure 3.7 shows a simple example of message passing step in a 3-node graph.

3.4 Attention Mechanisms

The *attention mechanism* [94] was first adopted in machine translation applications, and specifically to improve the performance of models that deal with long sequences of words. This method consists in calculating a weighted sum of a representation of the input, where the weight of each part of the input (such as the weight assigned to each word in a sentence to be translated) is based on how important this part of the input is when generating a given part of the output (such as a specific word in

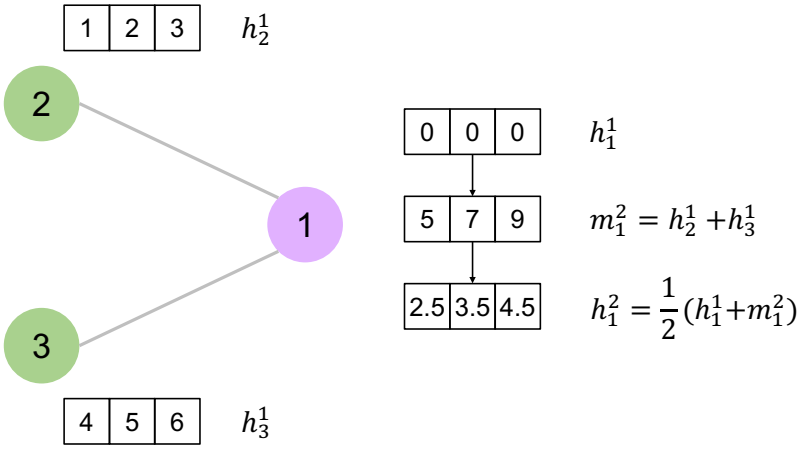


Figure 3.7 Simple example of message passing step for node 1, where the message function is a sum of the hidden states of the nearest neighbors and the update function is the mean of the message and the current hidden state

the translated sentence). For example, if the task consisted in translating the phrase *Il gatto ha sonno* to English (*The cat is sleepy*), the output word *cat* has a strong correlation with its direct translation *gatto*, and it's less dependent on the other words in the input, so *cat* should have a high weight when paired to *gatto*.

In practice, this is done via a database of key-value pairs $\mathcal{D} = \{(\mathbf{k}_1, \mathbf{v}_1), (\mathbf{k}_2, \mathbf{v}_2), ..., (\mathbf{k}_n, \mathbf{v}_n)\}$ and a *query* $\mathbf{q}$ [97]. The model will calculate a score (or attention weight) α between the query and each key $\mathbf{k}_i$ in the database via a scoring function, and then perform a pooling of the result via weighted average of the values $\mathbf{v}_i$ corresponding to each key (Figure 3.8).

$$\text{Attention}(\mathbf{q}, \mathcal{D}) = \sum_{i=1}^{n} \alpha(\mathbf{q}, \mathbf{k}_i)\mathbf{v}_i \tag{3.15}$$

In the example above, a vector representation of *cat* would be the query and the vector representations of each word in *Il gatto ha sonno* would be the keys and values. In this case, key and value are related to the same word, but their vector representation is not necessarily identical, for reasons of flexibility of the model. The weights need to be differentiable in order to be learned, and it also adds interpretability to the model if they are positive and sum up to 1, so it's common to calculate them by passing a scoring function $a(\mathbf{q}, \mathbf{k}_i)$ through a softmax function

$$\alpha(\mathbf{q}, \mathbf{k}_i) = \text{softmax}(a(\mathbf{q}, \mathbf{k}_i)) = \frac{\exp(a(\mathbf{q}, \mathbf{k}_i))}{\sum_j \exp(a(\mathbf{q}, \mathbf{k}_j))} \quad . \tag{3.16}$$

This mechanism is known as *soft* attention, as opposed to *hard* attention, where the weights are not differentiable and the gradient needs to be estimated via sampling.

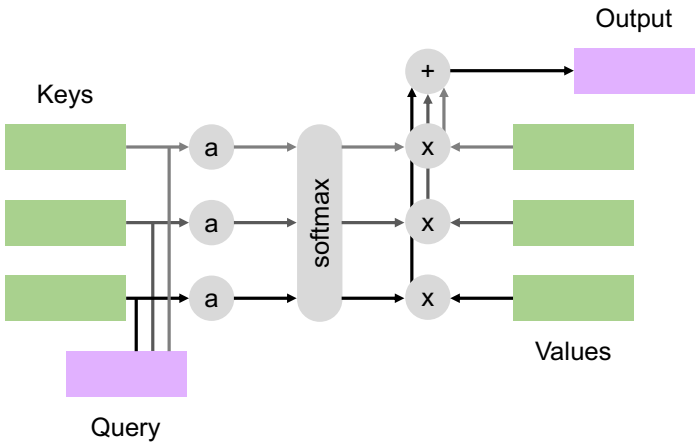


Figure 3.8 Schematic representation of the attention mechanism for one query. Adapted from Figure 11.3.1 in reference [94]

Some examples of scoring functions are

- Scaled dot product

$$a(\mathbf{q}, \mathbf{k}_i) = \frac{\mathbf{q}^\mathsf{T}\mathbf{k}_i}{\sqrt{d}} \quad , \tag{3.17}$$

  where d is the length of $\mathbf{q}$ and $\mathbf{k}_i$.

- Trainable weight matrix [98]

$$a(\mathbf{q}, \mathbf{k}_i) = \mathbf{q}^\mathsf{T}\mathbf{W}\mathbf{k}_i \quad . \tag{3.18}$$

- Additive attention [99]

$$a(\mathbf{q}, \mathbf{k}_i) = \mathbf{w}^\mathsf{T}\tanh(\mathbf{W}[\mathbf{q}; \mathbf{k}_i]) \quad , \tag{3.19}$$

  where $\mathbf{q}$ and $\mathbf{k}_i$ are concatenated and passed through a single layer fully connected neural network.

It is also possible to perform several attention pooling processes in parallel, with independently learned scores, to allow the model to form different representations at the same time. The results are then pooled by some operation like concatenation or sum or mean. This method is known as Multi-head attention.

Another variation of the attention mechanism is the so-called Self-attention or Intra-attention. In this case a representation of the input is created by computing the attention weights that relate different elements of the input to each other.

4 Generative Models

Generative models are a group of machine learning models that aim at learning the underlying distribution of available data or the construction principles of said data, in order to create new realistic elements that follow the same distribution or construction principles. These models have been applied in fields such as the generation of artificial images [100], text [101] and sound [102]. In computational chemistry, these models are valuable tools in the field of inverse design of novel materials and molecules with desired properties [103, 104]. In particular, they can be valid alternatives to high-throughput virtual screening and global optimization methods, because they allow the exploration of an extremely large chemical space, with minimal selection bias from the user [105, 106]. This approach has been adopted often in organic chemistry, because the space of organic molecules with certain desired properties is potentially infinite. The targeted generation of new materials can be done by implicitly learning features from a targeted database [107], by optimizing properties in a latent space [108–110], or by conditional generation.

In the field of inorganic materials, generative models need to face some additional challenges: while organic molecules are mostly made of a few chemical species and can be represented by strings such as SMILES [111] and SELFIES [112], or graph representations with well defined bonds [113], inorganic solids could be formed from almost any element in the periodic table, and have a crystalline structure that can't be inferred from the composition. Moreover, there is less availability and less diversity in the training data. Together with the usual requirements for materials representations in ML models, such as rotational and translational invariance and uniqueness, the material's representations for a generative model need to be invertible and possibly take periodicity into account [114]. Some examples of such representations are:

- Composition vectors have the atomic number of each element as one entry of the vector, and don't include any structural information. It's also possible to represent one element with multiple entries, such as the row and group number in the periodic table [115].

- Bag-of-atoms [116] also includes only the composition. Each compound is represented as a matrix that has all the possible elements as rows and the possible stoichiometric coefficients as columns, filled with 1 in correspondence of each (atom,stoichiometric coefficient) position and zero everywhere else (e.g SiO_2 will have a 1 in the positions (H,0),(He,0),(Li,0),...,(O,2),...,(Si,1),... and 0 everywhere else).

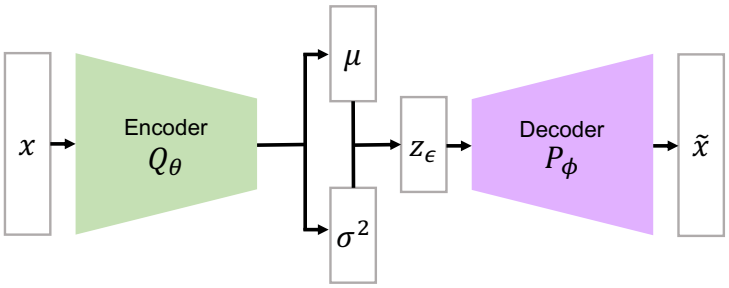


Figure 4.1 Variational Autoencoder

- Lattice parameters and element coordinates [117] such as in a poscar or cif file include also information about the structure in real space. Additional information about the momentum space can be added as well [118].
- 3D atomic density [119, 120] treats the structure as a three-dimensional grid of voxels, as in a 3D image
- Graphs [121] where the nodes represent atoms and the edges represent bonds.

In this context, deep generative models based on neural networks, have proven to be a very effective tool. Such models include Generative Adversarial Networks (GAN) [117, 122–126], Variational Autoencoders (VAE) [116, 118–120, 127–130], Recurrent Neural Networks (RNN), and Reinforcement Learning (RL) [131, 132].

4.1 Variational Autoencoder

Variational Autoencoders [133–135] are models that aim at learning the probability distribution of the data to then sample from it. However, only discrete samples from this distribution are available (i.e. the training set) and the full distribution is unknown. To solve this problem, a neural network can be used to sample from a known (and simpler) probability distribution, known as latent space, and learn how to map these points to a data-like space. In a VAE this network is called *decoder*, and its paired to a second neural network, the *encoder*, that learns how to map the initial training set to the latent space (Figure 4.1). The training of a VAE proceeds as follows. First the encoder Q_θ (with trainable parameters θ) maps the training set samples $\mathbf{x}_i$ to a multivariate normal distribution with mean μ and variance σ^2 in the latent space

$$q_\theta(\mathbf{z}|\mathbf{x}_i) = \mathcal{N}\left(\boldsymbol{\mu}(\mathbf{x}_i), \boldsymbol{\sigma^2}(\mathbf{x}_i)\right) \quad . \tag{4.1}$$

Mapping to a distribution instead of a single point allows to learn a more robust representation and a smooth latent space without regions that can't be decoded to valid

representations. Then the decoder P_ϕ (with trainable parameters ϕ) samples a point $\mathbf{z}_\epsilon$ from this distribution and tries to reconstruct the input $\mathbf{x}_i$. The weights and biases of the two networks are trained to minimize the following cost function

$$J(\mathbf{x}_i|\phi,\theta) = \underbrace{(\mathbf{x}_i - P_\phi(\mathbf{z}_\epsilon))^2}_{\text{reconstruction}} + \lambda \underbrace{D_{\text{KL}}\left(q_\theta(\mathbf{z}|\mathbf{x}_i) \;||\; \mathcal{N}(0,1)\right)}_{\text{regularization}}. \tag{4.2}$$

The first term allows the decoder to learn how to accurately reconstruct the training set, and it's simply the mean squared error between input and decoded output. The second term is a regularization term that drives the encoder to shape the latent space as a standard gaussian distribution $\mathcal{N}(0,1)$, and it's mathematically expressed as the Kullback-Leibler (KL) divergence between the encoder output and the target distribution. The minimum of the cost function is a trade-off between the two terms. If only the the reconstruction term was to be minimized, the decoder would accurately reproduce the training set but would not be able to generalize and generate new samples. Likewise, if only the regularization term was minimized, the decoder would not be able to create many distinct samples, and resort instead to output an average of the training set (a problem known as posterior collapse [136, 137]). The trade-off can be controlled via the hyperparameter λ. After training, the decoder part of the network can be used to sample from the latent space and generate new data

$$\tilde{\mathbf{x}} = P_\phi(\mathbf{z}) \quad . \tag{4.3}$$

In the case of conditional generation[116, 138–142], a label is appended to the input and then the same label is appended to the sampled elements that will be decoded. In this way the label affects how the encoder shapes the latent space, and how the decoder decodes the sample.

4.2 Generative Adversarial Network

The approach in a GAN [143] is similar to the VAE: a neural network, here called *generator* G_θ, samples from a pre-defined probability distribution, maps the samples to a data-like space and gets feedback form a second neural network to learn how to generate realistic samples. However, the training procedure is different. In the case of a GAN, the second neural network, the *discriminator* D_ϕ, receives real samples from the training set and fake samples from the generator, and tries to distinguish them (Figure 4.2). The two networks are trained in an adversarial fashion, so the discriminator must get better at distinguishing samples, while the generator must learn how to create more and more realistic samples. When the discriminator is not able to distinguish the two type of samples anymore, the training ends. At this point the fake generated samples should look exactly like samples coming from the real data distribution. However, the generator might resort to repropose always the same samples, that are known to be effective into fooling the discriminator. This problem is known as

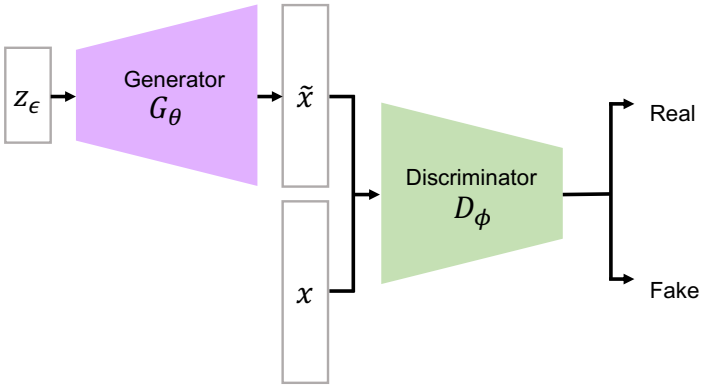


Figure 4.2 Generative Adversarial Network

mode-collapse [144–146], and has led to a series of different proposed cost functions and modified NN architectures to improve the training stability. One example is the Wasserstein GAN (WGAN) [147], where the discriminator outputs, instead of a simple real/fake binary value, a scalar value that measures the quality of the sample. In this case the discriminator is referred to as *critic*. The measure is the Wasserstein (also known as earth-mover) distance between real and generated samples, and it can be interpreted as the cost of moving along an optimal path the mass of one distribution to match the other. Given that the critic can output any value in the interval $(-\infty, +\infty)$, it is often beneficial to restrict the rate at which the output can change between two inputs by either clipping the weights of the critic, or adding a gradient penalty (GP) [148]. Including the gradient penalty term, the cost function of the WGAN-GP is

$$J_D(\mathbf{x}_i|\phi,\theta) = -\underbrace{D_\phi(\mathbf{x}_i)}_{\text{real}} + \underbrace{D_\phi(G_\theta(\mathbf{z}_\epsilon))}_{\text{fake}} + \underbrace{\lambda\mathcal{P}\left(\mathbf{x}_i, G_\theta(\mathbf{z}_\epsilon)\right)}_{\text{gradient penalty}} . \tag{4.4}$$

4.3 Reinforcement Learning

Reinforcement Learning (RL) is a machine learning model built to take a sequence of decisions that affect an evolving system, with the objective of achieving a pre-defined goal. Some examples of decision-making problems are the movement of robots [149], and playing games [150, 151]. More precisely, in RL the decisions are called *actions* and are taken by an *agent* that acts on an environment in a given *state* to maximize a *return*, i.e. the sum of the *rewards* that it receives after each action. After the action, the environment will transition to a new state. The specific form of the reward function is defined by the user, and depends on the application. The key feature of RL is that an action is affected by the actions previously taken by the model. This differs from most of the other deep learning models, where the prediction on some test data do not affect the predictions on other data successively presented to the model. Here

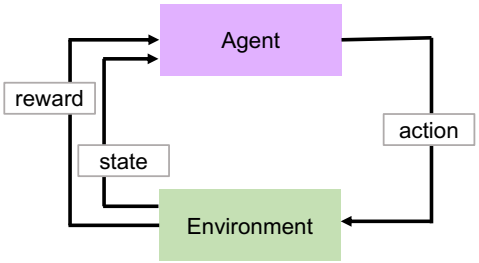


Figure 4.3 Reinforcement Learning

we denote the state at time t as $s_t \in S$, an action as $a_t \in A$, and the reward as $r_t = r(s_t, a_t)$. The return R is often calculated as a weighted sum of the rewards, where the weight is a discount factor γ that damps the values of the rewards the farther they are in time.

$$R = r_0 + \gamma r_1 + \gamma^2 r_2 + ... = \sum_{t=0}^{\infty} \gamma^t r_t \tag{4.5}$$

A high discount factor enables the exploration of the space by taking long sequences of actions, while a small value favors short sequences of actions. The agent is trained to take the best sequence of actions (i.e. the best *trajectory*) to maximize *R*. The probability of taking an action a given a state s is called a (stochastic) *policy* $\pi(a|s)$, with $\sum_a \pi(a|s) = 1$ for any state s. A deterministic policy is a special case of policy where only one action has a probability of 1 and all the others are 0. In practice, the policy corresponds to the algorithm that the agent uses to make decisions. RL problems normally rely on the assumption that the system is a Markov system, i.e. that the next state s_{t+1} only depends on the current state s_t and the action a_t. For a given policy π and an initial state s_0, we can define the action-value function

$$Q^{\pi}(s_0, a_0) = r(s_0, a_0) + \mathbb{E}_{a_t \sim \pi(a_t|s_t)} \left[\sum_{t=1}^{\infty} \gamma^t r_t \right] \tag{4.6}$$

as the reward for the first action plus the return averaged over all the possible future trajectories taken from s_0 according to π. The optimal policy π^* is the one that finds the maximum action-value function Q^*. Equation 4.6 can be also written in recursive form

$$Q^{\pi}(s, a) = r(s, a) + \gamma \sum_{s' \in S} P(s'|s, a) \sum_{a' \in A} \pi(a'|s') Q^{\pi}(s', a') \tag{4.7}$$

for all the states s and actions a. Here the successive action-value functions are weighted by the policy, i.e. by the probability of taking action a' given the state s' and by the transition rate P, i.e. the probability of reaching state s' given an initial state

s and an action a. P in most of the applications is not known *a priori*, but it's only accessible via sampling. One possible strategy to find an approximate $Q*$ without knowing P consists in choosing a policy π_e and using it to collect n trajectories with T time steps each, and minimize the loss function

$$J(Q) = \frac{1}{nT} \sum_{i=1}^{n} \sum_{t=0}^{T-1} \left[Q(s_t^i, a_t^i) - r(s_t^i, a_t^i) - \gamma \max_{a'} Q(s_{t+1}^i, a') \right]^2 \quad . \tag{4.8}$$

This approach is known as Q-learning. $J(Q)$ can be minimized via gradient descent to update Q in the following way

$$Q(s_t^i, a_t^i) \leftarrow Q(s_t^i, a_t^i) - \eta \nabla_Q J(Q) \tag{4.9}$$

The solution will converge to a value $\hat{Q}$ that approximates the optimal Q^*, as well as an approximate optimal policy $\hat{\pi} = \arg\max_a \hat{Q}(s, a)$. The initial choice of policy π_e is crucial to get samples that approximate well P. A simple choice could be

$$\pi_e(a|s) = \begin{cases} \arg\max_{a'} \hat{Q}(s, a'), & \text{with probability } p \\ \text{uniform}(A), & \text{with probability } 1 - p \end{cases} \tag{4.10}$$

This policy chooses the optimal action according to the current estimate $\hat{Q}$ with (user-defined) probability p and explores the environment by randomly sampling from all the possible actions in the other cases. Another example is the softmax exploration policy

$$\pi_e(a|s) = \frac{\exp\left(\beta \hat{Q}(s, a)\right)}{\sum_{a'} \exp\left(\beta \hat{Q}(s, a')/T\right)} \tag{4.11}$$

The hyperparameter β again controls how likely is the model to choose the optimal action (high β) or sample actions randomly ($\beta = 0$). When a neural network is used to estimate Q, the method is known as Deep Q-learning.

5 ML-based Screening of Halide Double Perovskites for Photovoltaic Applications

5.1 Motivation

The chemical space spanned by charge balanced double perovskites includes thousands of compositions, and many among those might have favorable optoelectronic properties. However, exploring this large space solely by DFT methods is computationally expensive, especially because this class of materials often can be accurately described only by hybrid functionals and including spin-orbit coupling effects [55]. High-throughput screening studies have been performed in the past [72, 73, 75, 77, 79, 81], but have been limited to only a fraction of the full chemical space. These studies employed DFT at GGA or hybrid level to compute compounds properties and their stability. Other works studied larger spaces of up to a few thousand double perovskites [82, 83], but employed geometric factors such as the Goldschmidt tolerance factor t, the octahedral factor μ or modified versions of those [48] to reduce the search space before the DFT calculations.

To allow the exploration of a larger chemical space, several groups employed machine learning-based methods to filter candidate materials. Among these, Schmidt et al. [152] first generated a database of almost 250,000 cubic perovskites calculated via DFT+U (641 of which were stable and 1562 with a gap above 0.5 eV), and then benchmarked the performance of several ML models (ridge regression, neural networks, random forest, and extremely randomized trees) on the prediction of the energy above hull . The most accurate method was found to be extremely randomized trees. Saidi et al. [153] used a hierarchical convolutional neural network to predict the lattice constant, octahedral angle and band gap of hybrid metal halide perovskites ABX_3, focusing on the effect of the organic cation A. Li et al. [154] pre-trained several ML models (Gradient Boosted Regression Tree (GBRT), Kernel Ridge Regression (KRR), support vector regression, bootstrap aggregating regression, Gaussian Process Regression (GPR) and random forest) on the formation energy of 1593 oxide single perovskites and used it to improve the prediction of the band gap. The lowest mean absolute error of 0.384 eV was achieved by the GPR model. Pilania et al. [155] applied KRR to predict the band gap of oxide double perovskites of the type AA'BB'O_6, starting from a database of 1306 DFT band gaps calculated with the GLLB-SC func-

tional [156]. An initial set of 16 elemental features and 16 LASSO-based [157] compound features were used to build descriptors to be tested via a Linear Least Square Fit (LLSF). The best-performing ones were used as input of the KRR model, with a final Root Mean Squared Error (RMSE) of 0.36 eV. Agiorgousis et al. [158] trained a random forest algorithm on the band gaps of chalcogenide double perovskites calculated via hybrid DFT using the HSE06 functional to identify promising solar absorbers. The stability of the selected compounds was evaluated by combining information from the Goldschmidt tolerance factor, the decomposition energy and molecular dynamics (MD) simulations. For the stable compounds, optical absorption was calculated, leading to the discovery of 5 promising sulfide double perovskites. In the field of halide double perovskites, Im et al. [159] trained a GBRT on PBE gaps of 540 structures, and predicted the formation energy and the band gap with a RMSE of 0.021 eV/atom and 0.223 eV respectively. Yang et al. [160] compared the performance of GBRT, ridge regression, support vector regression, KRR, a bagging ensemble algorithm, and a random forest ensemble algorithm and chose GBRT to explore an initial space of 16400 double perovskites. Konno [161] employed a convolutional neural network to extract elemental features from the position of the atoms in the periodic table. The CNN was trained and tested on 3734 experimental band gaps and predicted band gaps with a RMSE of 0.42 eV. This kind of Periodic Table Representation (PTR) allows to learn basic elemental properties and how they vary across the periodic table by just using the spatial arrangement of the atoms, as if they were pixels in an image. This has shown to be very useful situations where the elemental composition has the largest influence on the target property, as in the case of Heusler compounds [162] or, as in our case, double perovskites [163]. The advantage of the PTR is that no additional feature extraction procedure is necessary.

Here we use a CNN on the same kind of periodic table representation to sample a space of 7056 double perovskites ABC_2D_6 with 2 alkali metals in the position C, 44 metals in the position A/B and 4 halides in the position D (Figure 5.1). The focus is set on properties relevant for photovoltaic applications, such as the thermodynamic stability of the materials, their power conversion efficiency and the effective masses of their charge carriers.

5.2 Machine Learning Model

The input for the CNN is a tensor with 3 channels, one for each of the C, A/B, and D sites. Each channel is a grid in the shape of a periodic table with the stoichiometric coefficient (2 for C, 1 for A and B and 6 for D) in correspondence of the four elements, and zero everywhere else.

The neural network (cf. Table 5.1) has 5 convolutional layers with four 2x2 kernels and one 1x2 kernel, and 4 fully connected layers. Each layer has LeakyReLU [164]

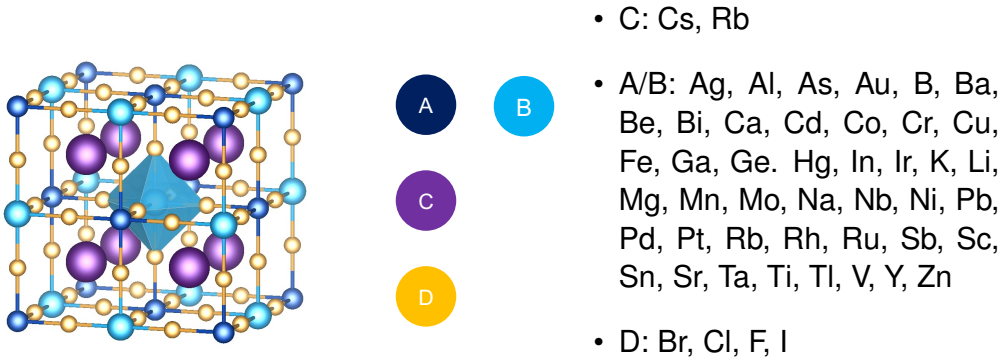


Figure 5.1 Elements on the lattice sites C, A/B and D

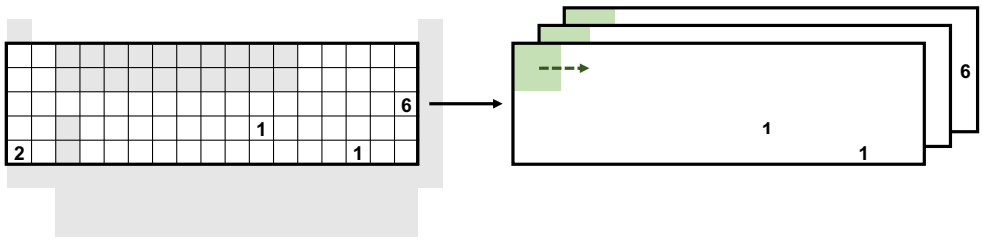


Figure 5.2 Periodic table representation with a 2x2 kernel, separated in 3 channels for A/B, C, and D.

as activation function, with a negative slope of 0.2. The fully connected layers include layer normalization [165] and dropout [166] with a probability of 0.25. The network was trained using the Adam optimizer [167] on batches of 100 samples for 1000 training set iterations, with a learning rate of 0.001, a weight decay of 0.0005 and cosine annealing [168] as learning rate scheduler. The hyperparameters were chosen in order to minimize the loss between predicted and calculated band gap values on the test set.

The training and test sets include 764 and 200 structures, respectively, calculated by DFT using the FHI-aims code [169]. We used the HSE06 [170, 171] exchange-correlation functional, a (4x4x4) k-point grid, an energy convergence threshold of 10^{-6} eV and a density convergence threshold of 10^{-6} e/a_0^3, where a_0 is a Bohr radius. We use the numeric atom-centered *light* basis sets as implemented in FHI-aims [169]. For most computations, the geometry was optimized in a symmetry-preserving frame-

Convolutional layers				
Kernel	Channels	Activation	Dropout	Norm
2x2	3,100	LeakyReLU(0.2)	0.25	-
2x2	100,100	LeakyReLU(0.2)	0.25	-
2x2	100,100	LeakyReLU(0.2)	0.25	-
2x2	100,100	LeakyReLU(0.2)	0.25	-
1x2	100,100	LeakyReLU(0.2)	0.25	-
Fully connected layers				
Nodes		Activation	Dropout	Norm
1200,200		LeakyReLU(0.2)	0.25	layer
200,200		LeakyReLU(0.2)	0.25	layer
200,200		LeakyReLU(0.2)	0.25	layer
200,1		LeakyReLU(0.2)	0.25	layer

Table 5.1 Network architecture

work, with a covergence threshold for the forces of 10^{-2} eV/Å. The calculations include a non-self-consistent spin-orbit coupling correction [172] and collinear treatment of the spin, which was initialized following the configuration of isolated atoms. In the structures containing two metals with unpaired electrons, the two spins were initialized in a antiparallel configuration.

For the structures with a predicted band gap between 0.9 and 1.6 eV, the the thermodynamic stability was estimated by calculating the enthalpy of the decomposition reaction [173]. For the stable compounds the absorption coefficient was calculated using the random phase approximation on an increased k-point grid density of (8x8x8) and a gaussian broadening of 0.05 eV. From this, the Spectroscopic Limited Maximum Efficiency (SLME) was calculated to estimate the maximum theoretical power conversion efficiency, following a method proposed by Yu and Zunger [174]. Additionally, the effective mass was calculated by parabolic fit of the band edges along the high symmetry directions.

5.3 Spectroscopic Limited Maximum Efficiency

In first approximation the efficiency of a solar cell can be calculated from the Schockley-Queisser (SQ) model, which considers an ideal solar absorber that has a step function as absorptivity, with a value of 1 above the band gap (perfect absorption) and 0 below the gap (no absorption at all). If we assume that every absorbed photon generates one electron-hole pair, then the short-circuit current density J_{SC}, i.e. the current that flows through the cell at zero voltage, can be calculated as

$$J_{SC} = e \int_{E_G}^{\infty} \phi_{sun}(E) dE \quad . \tag{5.1}$$

Here $\phi_{sun}(E)$ indicates the photon flux of the Sun per unit energy. Additionally, the cell will emit radiation with a spectrum that resembles that of a black body at the temperature of the cell. This generates a dark current density J_0

$$J_0 = e\pi \int_{E_G}^{\infty} \phi_{blackbody}(E, T_{cell}) dE \quad . \tag{5.2}$$

Overall, considering only the losses due to the photons below the gap and the radiative recombination, the current density J under illumination will be

$$J = J_{SC} - J_0 \left[\exp\left(\frac{eV}{k_B T}\right) - 1 \right] \quad . \tag{5.3}$$

The maximum efficiency η obtained by such device is the ratio between the maximum of the cell power density $J \cdot V$ and the incoming power density of the Sun

$$\eta = \frac{P_{MAX}}{P_{IN}} = \frac{max(J \cdot V)}{\int_0^{\infty} E\phi_{sun}(E) dE} \tag{5.4}$$

This shows that the optimum gap for a perfect absorber whould be 1.34 eV, leading

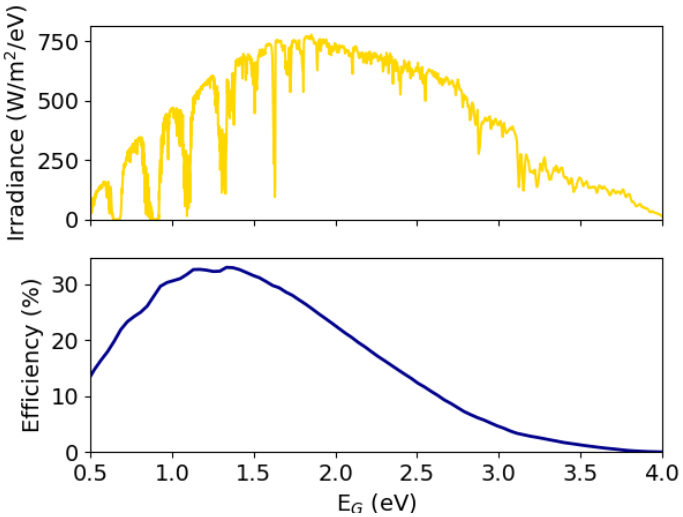


Figure 5.3 Irradiance of a standard solar spectrum [175] and power conversion efficiency calculated using the Schockely-Queisser model. The maximum of PCE is at 1.34 eV. The data for the solar spectrum are publicly available free of charge under the following conditions: https://www.nrel.gov/disclaimer.html.

to an efficiency of 34%. A second approximation to approach more realistic values consists in including the absorptivity A of the material instead of a step function. In this case J_{SC} can be calculated by integrating over the solar spectrum from 0 to infinity, but with A as weighting factor. This could be already a good approximation for a direct

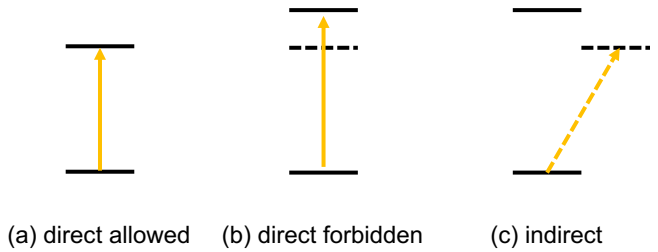


Figure 5.4 Three types of semiconductors based on the nature of their band gap: (a) direct gap with dipole-allowed transition, (b) direct gap with dipole-forbidden transition, and (c) indirect gap

gap semiconductor, however for the majority of the materials there are more loss mechanisms to take into account. Figure 5.4 shows different types of semiconductors based on the nature of their band gap. Figure 5.4a corresponds to a material with a direct band gap, and a dipole allowed (DA) transition in correspondence of E_G. This is the situation that is most similar to the ideal case. Figure 5.4b is a direct gap semiconductor, but the first dipole-allowed transition happens at an energy higher than the band gap. This means that part of the energy absorbed will be released when the excited carriers relax to the band edge, and will not contribute to the power that can be produced by the device. The last case is an indirect gap semiconductor, and also in this case the first direct dipole-allowed transition will happen at a higher energy, and lead to some losses. Following an approach suggested by Yu and Zunger [174], these losses can be modeled as a Boltzmann factor $f_r = \exp\left(-\frac{E_{DA}-E_G}{k_B T}\right)$, that decays when the first dipole-allowed transition is very far from the energy gap E_G. This is based on the assumption that electron concentrations at the different energy levels follow a Boltzmann statistic and are the only factor that determines the ratio of radiative to non-radiative recombination. A different method proposed by Blank et al. [176] includes the non-radiative recombination in the model by introducing as a parameter the internal photoluminescence quantum yield (Q_i). Q_i is the number of carriers collected at the cell divided by the number of photons that are actually absorbed by the cell, so it's corrected for the losses due to reflection.

With the inclusion of this factor, the current density follows

$$\begin{aligned} J_{SC} &= e \int_0^\infty A(E)\phi_{sun}(E)dE \\ J_0 &= \frac{e\pi}{f_r} \int_0^\infty A(E)\phi_{blackbody}(E)dE \\ J &= J_{SC} - J_0 \left[\exp\left(\frac{eV}{k_B T}\right) - 1\right] \end{aligned} \tag{5.5}$$

and the efficiency can be calculated as in Equation 5.4. In this case the power conversion efficiency is known as Spectroscopic Limited Maximum Efficiency (SLME). The absorptivity can be calculated from the absorption coefficient α and a simple exponential decay.

$$A(E) = 1 - \exp\left(-2\alpha(E)L\right) \tag{5.6}$$

The absorption coefficient for the component i (x,y, or z) α_i can be obtain from Kohn-Sham states from

$$\alpha_i = \frac{8\pi^2}{\omega V_{cell}} \sum_{c,v} \sum_{\mathbf{k}} |p_{i;c,v,\mathbf{k}}|^2 \delta(\epsilon_{v,\mathbf{k}} - \epsilon_{c,\mathbf{k}} - \omega) d\mathbf{k} \quad , \tag{5.7}$$

where p indicates the momentum. When the transition is symmetry-forbidden, these matrix elements will be zero.

5.4 Results

The result of the training on a randomly sampled set of 200 structures is shown in Figure 5.5. With the exception of some outliers the model can predict the band gap with a reasonable accuracy and an overall MAE of 0.21 eV and RMSE of 0.45 eV. The RMSE here is higher than the MAE because it is more sensitive to the presence of outliers. The machine learning model is then used to select candidates with a predicted band gap between 0.9 and 1.6 eV. From this set of 459 structures, first those that contain toxic elements have been excluded. Subsequently, the band gap of the remaining 303 structures was explicitly computed at the hybrid DFT level (unless they were already contained in the initial training or test set for the ML model). For the 119 compounds with a DFT band gap included in the same interval, the decomposition enthalpy has been calculated (cf. Figure 5.6). For a compound to be stable the decomposition enthalpy must be positive, but given the finite accuracy of the employed approximate DFT functional and the fact that the DFT calculations don't include any temperature effect, also the structures with a decomposition enthalpy between -50 and 0 meV/atom have been included in the successive analysis - as candidates potentially stable at room temperature or metastable. As it has been shown by Sun et al. [177], metastable compounds are often found in this energy interval.

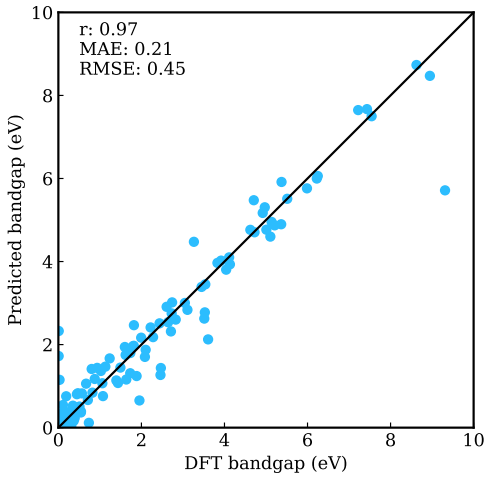


Figure 5.5 Training result on a test set of 200 structures

The maximum theoretical power conversion efficiency of the 12 stable and 17 metastable candidates thus identified was calculated for a layer thickness up to 5 μm (see Figure 5.7).

The results are summarized in Table 5.2.

Efficiencies above 20% were found for compounds with a dipole-allowed transition in correspondence with their direct or quasi-direct band gap. This is the case for $GeSnCs_2Cl_6$ which has a direct gap of 1.41 eV and and efficiency of 28%, for $GeSnCs_2Br_6$, which has a direct gap of 0.95 eV and an efficiency of 23%, and for $CrInRb_2I_6$, with a direct gap at 0.97 eV and a transition at 1.00 eV. $GeSnCs_2Cl_6$ and $GeSnCs_2Br_6$ also show optimal charge transport properties due to their low and balanced carrier effective masses (-0.13m_0 and 0.20m_0 for $GeSnCs_2Cl_6$, -0.09m_0 and 0.14m_0 for $GeSnCs_2Br_6$, in units of electron masses m_0 for holes and electrons, respectively).

$CrInRb_2I_6$ has heavier and anisotropic effective masses, but still comparable to that of $AgBiCs_2Br_6$. Also some materials with indirect gap show a high efficiency, namely $AlLiCs_2I_6$, $CuInCs_2Cl_6$, $AgInCs_2Br_6$,$CrGaRb_2I_6$ and $CrGaCs_2I_6$. However these compounds have negative decomposition enthalpy and might not be stable at room temperature. Additionally, their bands are definitely narrower.

Among the structures with positive decomposition enthalpy , the highest efficiency (17%) was found for $CrScCs_2I_6$, with an indirect gap of 1.37 eV and absorption energy of 1.88 eV. The carriers effective masses are of the order of 0.5 m_0, except for holes along the L-Γ direction, were m_h^* is -0.87m_0.

Finally non symmetry-constrained relaxations show that with a threshold of 0.1 Å the majority of the structures retains the initial cubic spacegroup *Fm-3m* with the exception of 4 tetragonal structures, namely $AgIrRb_2Br_6$ (*I4*), $CuInCs_2Cl_6$ (*I4/mmm*), $CuFeRb_2Cl_6$ (*I4/mmm*) and $CuRuRb_2F_6$ (*I4/mmm*), one monoclinic structure, $FeRbCs_2I_6$ (*C2/m*), and one triclinic, $AuMoRb_2Br_6$ (*P-1*).

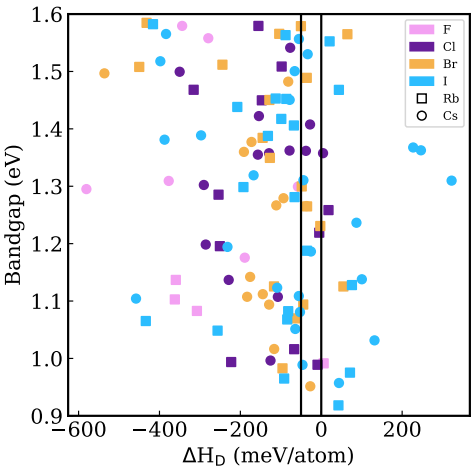


Figure 5.6 Enthaply of the decomposition reaction as a function of DFT band gap. The compounds with $\Delta H_D > 0$ meV/atom are stable, while those included between the vertical lines (from -50 to 0 meV/atom) are assumed to be metastable (see text)

Due to this second relaxation, the decomposition enthalpy of $AgInRb_2Br_6$ reaches -0.7 meV/atom and a power conversion efficiency of 17%.

5.5 Conclusions

In this project, we screened a large chemical space (7056 compounds) of inorganic halide double perovskites to uncover suitable candidates for photovoltaic applications. We applied a funnel-type approach to identify a pool of potential candidates and then reduce it by successively performing more demanding calculations based on band gap, thermodynamic stability, power conversion efficiency and carrier effective masses. Thereby we employed a state-of-the-art ML approach as a first step to limit the number of expensive band-structure calculations to just the 964 compounds used in training and testing the ML model. This is based on a neural network architecture composed of convolutional and fully connected layers with a periodic table representation of the perovskites. This approach yielded a high accuracy for the prediction of band gaps versus DFT results. The latter were all computed using high accuracy hybrid DFT including spin-orbit coupling in order to ensure high predictivity of our results. We find a number of very high performing compounds—with efficiencies as high as 28% and very low carrier effective masses (-0.13m_0 for holes and 0.20m_0 for electrons) for $GeSnCs_2Cl_6$. Unfortunately, our calculations show that such high performing compounds might only be meta-stable. Among the compounds predicted to be thermodynamically stable, we still find some with efficiencies of up to 17% ($AgInRb_2Br_6$) albeit with worse and more anisotropic effective masses. Notably, when

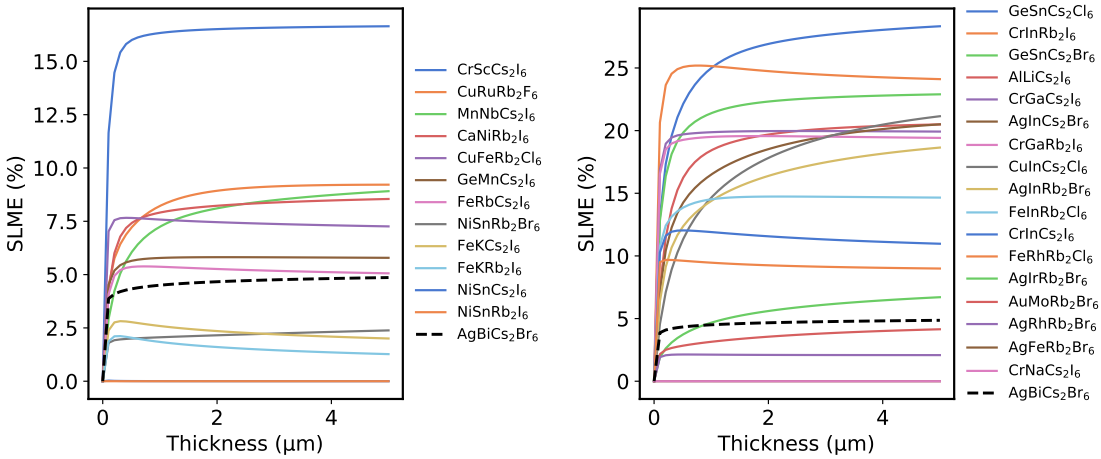


Figure 5.7 SLME for stable (left) and metastable (right) structures. The efficiency of $AgBiCs_2Br_6$ is shown for comparison

relaxing the strict requirement of cubic symmetry, we find 6 compounds to achieve higher stabilities at lower symmetries.

Thus, while we do find a few novel materials, trade-offs between power conversion efficiency, carrier mobility and (meta-)stability may indeed be unavoidable for this materials class.

Structure	ΔH_D (meV/at)	E_{gap} (eV)	E_{abs} (eV)	SLME (%)	m_h^* (m_0)	m_e^* (m_0)
$AgIrRb_2Br_6$	-48	1.30	2.03	6	-1.71 (X-Γ) -3.81 (X-W)	0.34 (X-Γ) 0.31 (X-W)
$CrNaCs_2I_6$	-46	0.99	2.20	0	-1.88 (Γ-X) -0.95 (Γ-L)	0.49 (X-Γ) 0.52 (X-W)
$AgFeRb_2Br_6$	-45	1.09	2.31	0	-0.38 (Γ-X) -0.52 (Γ-L)	0.56 (X-Γ) 0.46 (X-W)
$AlLiCs_2I_6$	-44	1.31	1.67	20	-0.81 (Γ-X) -1.24 (Γ-L)	0.28 (Γ-X) 0.28 (Γ-L)
$CrInRb_2I_6$	-43	0.97	1.00	24	-0.42 (L-Γ) -0.14 (L-W)	0.65 (L-Γ) 0.19 (L-W)
$CuInCs_2Cl_6$	-38	1.36	1.65	19	* (Γ-X) -1.06 (Γ-L)	0.27 (Γ-X) 0.27 (Γ-L)
$AgInCs_2Br_6$	-36	1.27	1.61	20	-5.62 (Γ-X) -0.71 (Γ-L)	0.20 (Γ-X) 0.20 (Γ-L)
$AgRhRb_2Br_6$	-36	1.49	2.67	2	-1.06 (L-Γ) -0.69 (L-W)	0.36 (X-Γ) 0.30 (X-W)
$CrGaRb_2I_6$	-35	1.19	1.49	20	-1.26 (Γ-X) -0.86 (Γ-L)	0.66 (L-Γ) 0.26 (L-W)
$AuMoRb_2Br_6$	-35	1.26	2.15	4	*	*
$CrInCs_2I_6$	-29	1.03	1.38	11	-0.45 (L-Γ) -0.15 (L-Γ)	0.66 (L-W) 0.21 (L-W)
$GeSnCs_2Cl_6$	-28	1.41	1.41	28	-0.13 (Γ-X) -0.13 (Γ-L)	0.20 (Γ-X) 0.20 (Γ-L)
$GeSnCs_2Br_6$	-27	0.95	0.95	23	-0.09 (Γ-X) -0.09 (Γ-L)	0.14 (Γ-X) 0.14 (Γ-L)
$CrGaCs_2I_6$	-26	1.19	1.49	20	-1.22 (Γ-X) -0.88 (Γ-L)	0.67 (L-Γ) 0.27 (L-W)
$FeRhRb_2Cl_6$	-9	0.99	1.58	9	-0.39 (L-Γ) -0.36 (L-W)	1.80 (X-Γ) 0.59 (X-W)
$FeInRb_2Cl_6$	-4	1.19	1.73	15	-0.29 (X-Γ) -1.35 (X-W)	27.30 (L-Γ) 8.31 (L-W)
$AgInRb_2Br_6$	-2	1.23	1.58	17	-7.39 (Γ-X) -0.66 (Γ-L)	0.20 (Γ-X) 0.20 (Γ-L)
$CuRuRb_2F_6$	6	0.99	1.44	9	-1.15 (Γ-X) -1.59 (Γ-L)	15.00 (L-Γ) 7.60 (L-W)
$CuFeRb_2Cl_6$	18	1.26	2.05	7	* (Γ-X) -0.79 (Γ-L)	0.78 (X-Γ) 0.62 (X-W)
$NiSnRb_2I_6$	42	0.92	1.96	0	-0.20 (X-Γ) -0.99 (X-W)	0.80 (L-Γ) 0.40 (L-W)
$CaNiRb_2I_6$	44	1.47	2.33	8	-1.20 (Γ-X) -0.85 (Γ-L)	0.69 (X-Γ) 0.92 (X-W)
$NiSnCs_2I_6$	44	0.96	1.93	0	-0.21 (X-Γ) -1.09 (X-W)	0.82 (L-Γ) 0.40 (L-W)
$NiSnRb_2Br_6$	65	1.56	2.72	2	-0.22 (X-Γ) -1.18 (X-W)	0.99 (L-Γ) 0.50 (L-W)
$FeKRb_2I_6$	76	1.13	2.13	1	-0.17 (Γ-X) -1.14 (Γ-L)	0.55 (X-Γ) 0.80 (X-W)
$CrScCs_2I_6$	77	1.37	1.88	17	-0.87 (L-Γ) -0.55 (L-W)	0.54 (X-Γ) 0.46 (X-W)
$FeRbCs_2I_6$	87	1.24	2.13	5	-1.97 (Γ-X) -1.46 (Γ-L)	8.25 (X-Γ) 1.17 (X-W)
$FeKCs_2I_6$	100	1.14	2.10	2	-1.56 (Γ-X) -1.21 (Γ-L)	5.72 (X-Γ) 1.63 (X-W)
$MnNbCs_2I_6$	246	1.36	2.16	8	*	*
$GeMnCs_2I_6$	322	1.31	2.43	6	-0.22 (X-Γ) -0.87 (X-W)	0.27 (L-Γ) 0.20 (L-W)

Table 5.2 Decomposition enthalpy, DFT band gap, absorption energy, spectroscopic limited maximum power conversion efficiency at 5μm and carrier effective masses of the identified metastable and stable compounds, after symmetry constrained geometry optimization. Missing values (*) are due to one or both band edges having bands too narrow to calculate the curvature.

Structure	ΔH_D (meV/atom)	E_{gap} (eV)	E_{abs} (eV)	SLME (%)	Spacegroup symbol
$AgIrRb_2Br_6$	-32	1.95	2.43	15	I4
$CrNaCs_2I_6$	-44	0.99	1.63	8	Fm-3m
$AgFeRb_2Br_6$	-43	1.04	2.31	0	Fm-3m
$AlLiCs_2I_6$	-42	1.30	1.67	20	Fm-3m
$CrInRb_2I_6$	-42	0.91	0.93	23	Fm-3m
$CuInCs_2Cl_6$	-28	2.21	2.54	12	I4/mmm
$AgInCs_2Br_6$	-34	1.22	1.58	18	Fm-3m
$AgRhRb_2Br_6$	-33	1.45	2.27	8	Fm-3m
$CrGaRb_2I_6$	-34	1.19	1.47	19	Fm-3m
$AuMoRb_2Br_6$	6	2.69	3.27	4	P-1
$CrInCs_2I_6$	-28	0.97	1.00	24	Fm-3m
$GeSnCs_2Cl_6$	-27	1.35	1.35	27	Fm-3m
$GeSnCs_2Br_6$	-26	0.89	0.90	21	Fm-3m
$CrGaCs_2I_6$	-24	1.21	1.48	21	Fm-3m
$FeRhRb_2Cl_6$	-8	0.93	1.54	10	Fm-3m
$AgInRb_2Br_6$	-0.7	1.20	1.57	17	Fm-3m
$FeInRb_2Cl_6$	-4	1.06	1.61	14	Fm-3m
$CuRuRb_2F_6$	7	0.91	1.40	6	I4/mmm
$CuFeRb_2Cl_6$	19	1.13	2.10	4	I4/mmm
$NiSnRb_2I_6$	43	0.87	1.97	0	Fm-3m
$CaNiRb_2I_6$	44	1.48	2.35	8	Fm-3m
$NiSnCs_2I_6$	45	0.92	1.94	0	Fm-3m
$NiSnRb_2Br_6$	66	1.51	2.72	1	Fm-3m
$FeKRb_2I_6$	77	1.13	2.15	1	Fm-3m
$CrScCs_2I_6$	79	1.37	1.88	17	Fm-3m
$FeRbCs_2I_6$	105	1.40	2.13	11	C2/m
$FeKCs_2I_6$	102	1.14	2.13	2	Fm-3m
$MnNbCs_2I_6$	248	1.41	2.17	9	Fm-3m
$GeMnCs_2I_6$	323	1.29	2.43	6	Fm-3m

Table 5.3 Decomposition enthalpy, DFT band gap, absorption energy, spectroscopic limited maximum power conversion efficiency at 5μm of the metastable and stable compounds, after non symmetry constrained geometry optimization

6 Interpretable Band Gap Prediction of Double Perovskites

6.1 Motivation

The direct mapping of materials compositions to properties via high throughput virtual screening or machine learning methods has proven to be very effective for materials discovery applications. However, machine learning methods often lack interpretability, especially in the case of neural networks, where many parameters are needed but the parameters themselves do not provide physically relevant information on how a compound is related to its corresponding property. In this regard, empirical and semi-empirical methods have been used to gain insight in the material-property relation in fields such as semiconductors, electrochemistry and catalysis. One example in the context of band gap prediction of semiconductors is the Solid State Energy (SSE) scale developed by Pelatt et al. [178]. In this study, experimental values of ionization potentials and band gaps of 69 closed-shell binary compounds where used to define the position of atomic energy levels in a solid, similarly to standard reduction potentials in a solution of electrochemistry. The SSE of an element is then defined as the average electron affinity (for a cation) or average ionization potential (for an anion) of all the compounds containing said element. The electron affinities where calculated as the difference between ionization energy and energy gap. The trend of ionization potentials and electron affinities shows that these value converge around 4.5 eV below the vacuum level, which is the same value as the hydrogen donor/acceptor ionization energy $\varepsilon(+/-)$ [179], so that elements with SSE above $\varepsilon(+/-)$ will behave as cations and elements below will behave as anions. The SSE scale initially included 40 elements, and was successively extended to include 94 elements in different oxidation states [180]. A following study by Davies et al. [181] used this model to estimate the band gap of inorganic materials as

$$E_{gap}^{SSE} = SSE_{cation} - SSE_{anion} \tag{6.1}$$

for binary compounds, and as

$$E_{gap}^{SSE} = \min(SSE_{cations}) - \max(SSE_{anions}) \tag{6.2}$$

for materials with more than two elements. This method was tested on 35 ternary semiconductors and leads to a root mean squared error of 0.66 eV.

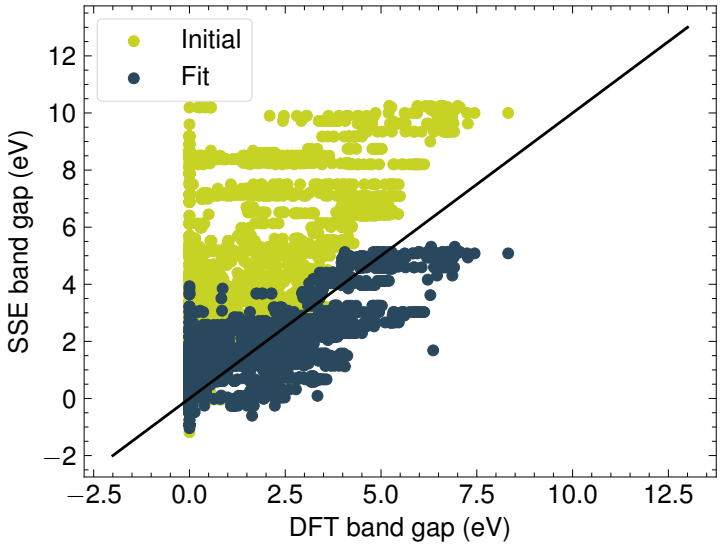


Figure 6.1 Band gap calculated from SSE values. The initial values tabulated in reference [178] were fitted again on data from the materials project to adjust for DFT-calculated band gaps

Following this example, we estimated the band gap of double perovskites from SSE values. Then, to allow more flexibility, we adopted a model to predict two electronic energy levels per element (one in the valence band and one in the conduction band) and calculated the band gaps from these. To add to the explainability of the model, we adopt a neural network architecture based on the model *Roost* (Representation learning from stoichiometry) [182], a message passing neural network with a soft self-attention mechanism. The self-attention assigns a weight to pair-wise element interactions and highlights which interactions have a higher contribution to the final property. This kind mechanism has been adopted also in similar architectures such as *CrabNet* [183], *Finder* [184] and other Transformer-based models [185].

6.2 Solid State Energy of Double Perovskites

Following the work of Davies et al. [181], we calculated the band gap of double perovskites from the Materials Project Database [186–190] using the SSE experimental values from reference [178] as initial values. Because of the 4 lattice sites in the structure ABC_2D_6, we chose the lowest among the cations SSE and the highest among the anions SSE (6.2). To correct for the difference between experimental values used in the original SSE values and the PBE band gaps of the Materials Project, we fitted again the values using gradient descent (Figure 6.1). Because of the use of only two values, the model can only predict one band gap value for all the structures that have

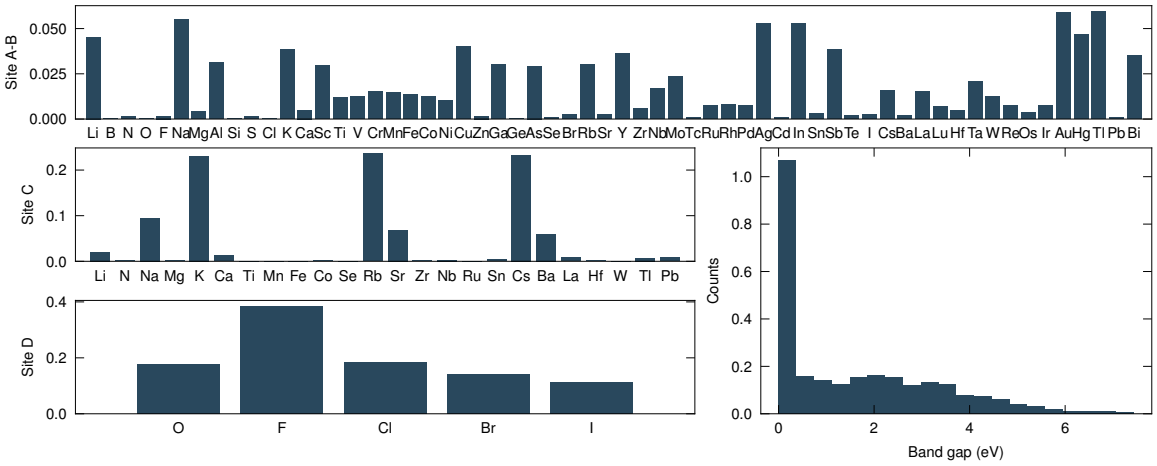


Figure 6.2 Element distribution in the training set normalized by site and band gap distribution

the same two elements with lowest (for cations) and highest (for anions) SSE. Moreover, the solid state energy is defined only as one value per element, and allows only a very simple description of the band gap.

6.3 Materials Representation and Model Architecture

To get a more complex picture of the effect of each element on the band gap, we adopted a message passing neural network trained on a dataset of 1941 cubic double perovskites with chemical formula ABC_2D_6 from the Materials Project Database, with their respective DFT band gaps. This set was divided in 1500/220/221 randomly sampled structures for training/validation test sets. The input structures are represented as 4 one-hot encoded vectors, one for each of the 4 sites. Also in this case the training set was expanded to include all the permutations BAC_2D_6, yielding a total of 3000 structures. The element distribution in the training set, normalized by site, and the band gap distribution are shown in Figure 6.2 The neural network architecture is represented in Figure 6.3, for a binary compound AB with two one-hot encoded input vectors.

The forward pass proceeds as follows:

1) The one-hot encoded atomic feature vectors are embedded into a to 63-dimensional space.

2) The fractional weight of each atom is appended to the corresponding feature vector as 64th dimension.

3) The 64-dimensional feature vectors are concatenated to form all the 16 pairs of the type A-A, A-B, A-C, A-D, B-A, B-B, ..., D-A, ..., D-D.

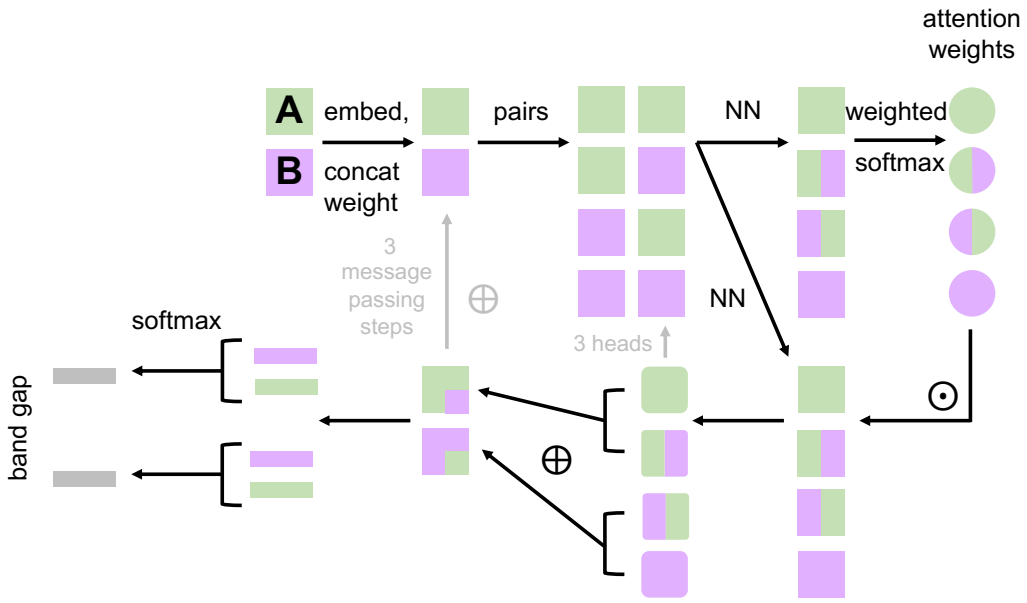


Figure 6.3 Example of network architecture with a binary compound AB

4a) The 128-dimensional pairs are passed through a single hidden layer neural network with a one-dimensional output, and successively through a weighted softmax function, with weights corresponding to the fractional weights of the elements in the structure, to obtain normalized *attention weights*.

4b) The 128-dimensional pairs obtained at step 3 are passed through a second single hidden layer neural network with a 64-dimensional output.

5) The feature vectors obtained in step 4b are multiplied to the corresponding attention weights obtained at point 4a, to form weighted feature vectors.

6) The steps 4a, 4b, and 5 are repeated three times in parallel (i.e. three *attention* heads) and the results are averaged, then summed over the first element of the pair.

7) The new 64-dimensional atomic feature vector is added to the previous one (step 2) and then steps 3-7 are repeated three times (three message passing layers).

8) Two independent linear layers are used to predict two energy levels per atom, one in the valence band and one in the conduction band.

9) The valence band maximum and conduction band minimum are calculated as weighted sums of the individual levels, with weights from a softmax function. In this way the highest contribution to the position of the band edges comes

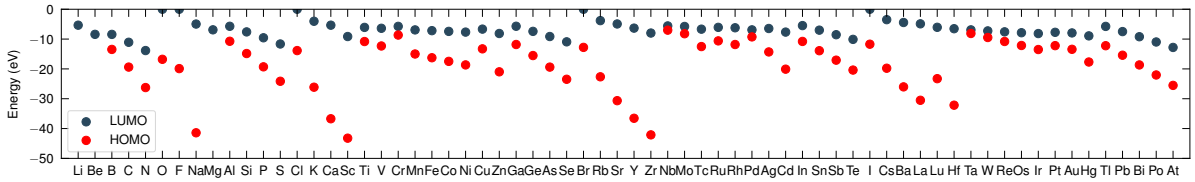


Figure 6.4 Reference values used for training the network

from the atom with the highest among the occupied levels and the atom with the lowest among the unoccupied levels, with an exponentially decreasing weight from the other atoms.

10) The gap is calculated from the difference of the band edges.

$$E_{gap} = \sum_{i}^{N_{atoms}} w_i \mathrm{LUMO}_i - \sum_{i}^{N_{atoms}} w_i \mathrm{HOMO}_i \tag{6.3}$$

All the single hidden layer neural networks in the model have 250 nodes and a LeakyReLU activation function, with a negative slope of 0.2. The architecture and hyperparameters were the same adopted in the *Roost* model [182], however the model was trained for 1000 training set iterations, to minimize the error the validation set.

To establish an initial ordering of the atomic energy levels we set as reference the atomic energies from Desclaux [191], using their most common oxidation state in the dataset. Because the empty s states of the anions were not available, these values were initialized as 0, and assumed to give a negligible contribution to the calculation of the gap. Here we will refer to the highest occupied and lowest unoccupied atomic energy levels as HOMO and LUMO, respectively. The reference values are shown in Figure 6.4. The loss function to train the network includes two terms: the first is the mean squared error between the predicted gap and the corresponding training value, the second term is a penalty term that keeps the predicted energy levels close to their corresponding reference

$$\begin{aligned} J = &\left(E_{gap} - E_{gap}^{train}\right)^2 \\ &+ \lambda \left[\sum_{i}^{N_{atoms}} \left(\mathrm{LUMO}_i - \mathrm{LUMO}_i^{ref}\right)^2 \right. \\ &\left. + \sum_{i}^{N_{atoms}} \left(\mathrm{HOMO}_i - \mathrm{HOMO}_i^{ref}\right)^2 \right] \end{aligned} \tag{6.4}$$

To establish the weight λ of the penalty term, we compared the results of the fit on the same model trained with and without prior. Figure 6.5 shows that introducing a prior does not significantly reduce the model performance. The chosen value for λ was therefore 0.001. Figure 6.6 shows the result of the band gap fit on a test set of

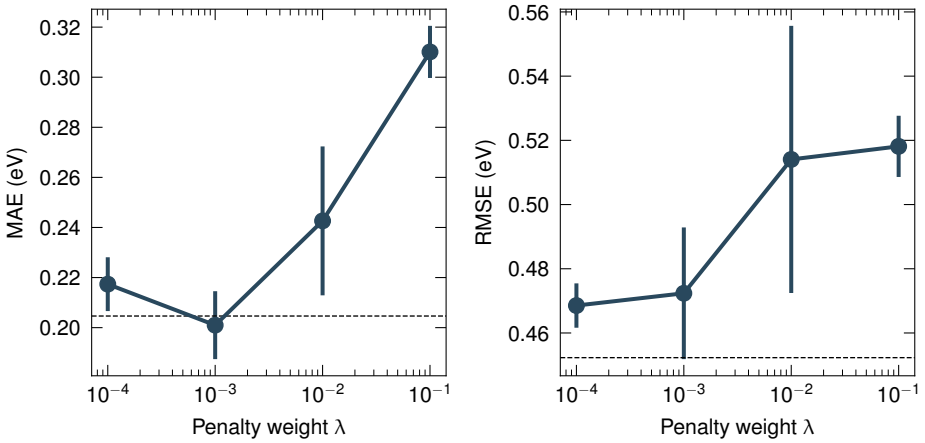


Figure 6.5 Average MAE and RMSE over 5 model initializations on a validation set of 220 structures, with standard deviation as error bar, for 4 values of λ. The dashed lines represent the corresponding average error of the model trained without prior.

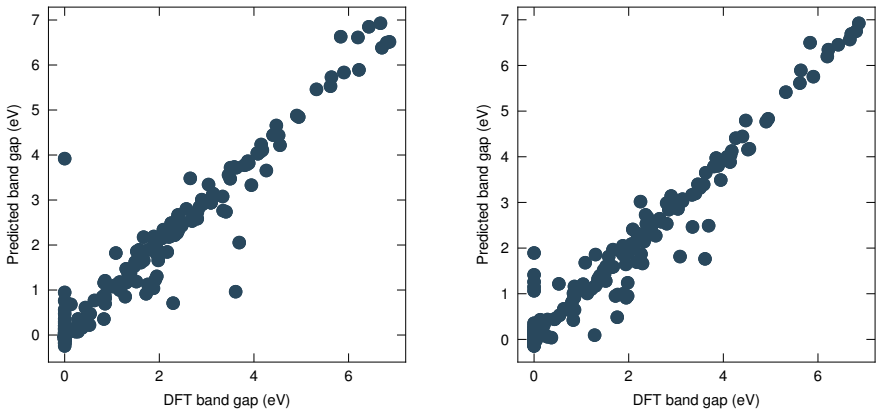


Figure 6.6 Model fit on a test set of 221 structures, without prior (left) and with a penalty weight of 0.001 (right)

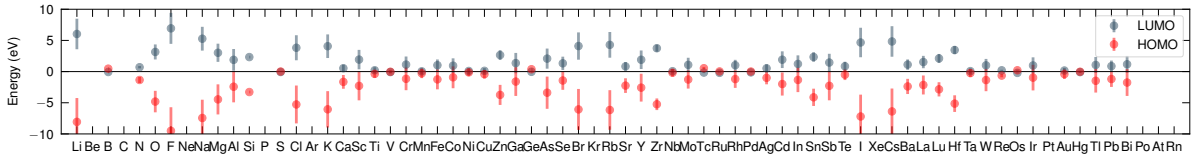


Figure 6.7 Predicted HOMO and LUMO values for a madel trained without prior, averaged over different structures, with standard deviation as error bar.

221 structures for a λ value of 0.001. In the following the parameters of the network will be analyzed, first for the model trained without initial references, the for the model with prior.

6.4 Training Without Prior

The average predicted values for HOMO and LUMO of each element are shown in Figure 6.7. Without providing the model with an initial reference, the values are symmetric respect to zero. From this we can see that the network will place transition metals close around the zero, in a way that on average they will give a high contribution to the calculation of the gap, and on average form structures with low band gaps. Other metals and anions are farther away, so their contribution will be smaller. However some reasonable trends also appear, for example despite the low contribution, alkali metals such as Li, Na, K, Rb, and Cs will lead to progressively smaller gaps, and in the same way F will on average give larger gaps that the other halides.

To analyze the parameters of the network we concatenated the attention weights averaged over the three attention heads with the softmax weights used to calculate the position of the band gap, and run a Principal Component Analysis (PCA) to reduce the dimensionality of the obtained feature vectors. We found that with 3 dimensions the explained variance of the data is 96% (80%, 12%, and 4% respectively). Successively we clustered the transformed data using the k-means clustering algorithm. This method allows to divide a set of vectors into k clusters C with centroids μ, by minimizing the Within-Cluster Sum-of-Squares Criterion (WCSSC):

$$\sum_{i=0}^{n} \min_{\mu_j \in C} \left(||x_i - \mu_j||^2 \right) \quad . \tag{6.5}$$

The k-means clustering algorithm requires the number of clusters as an input, therefore we repeated the clustering with different numbers of clusters and calculated the respective silhouette score

$$s(i) = \frac{b(i) - a(i)}{\max\{b(i), a(i)\}} \quad . \tag{6.6}$$

Here $a(i)$ is the average distance between each datapoint i and all the other points within the cluster to which i belongs, and $b(i)$ is the average distance between point

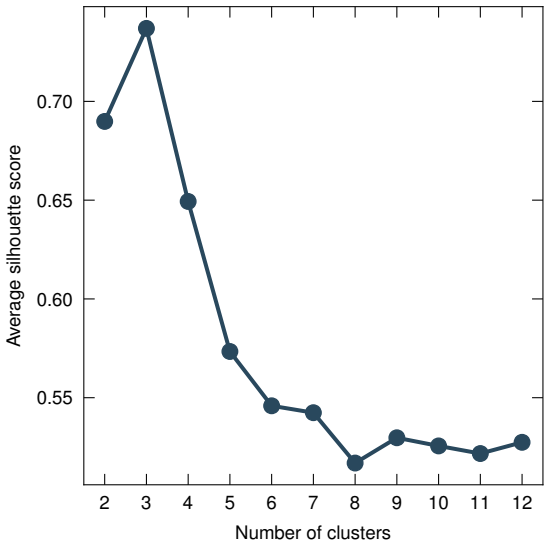


Figure 6.8 Average silhouette score for a different number of clusters

i and all the points in the closest cluster to which i does not belong. The silhouette score can vary between -1 and 1. Its value approaches 1 when the clusters are very well separated, 0 when the point i is at the edge between two overlapping clusters, and -1 if the point has been assigned the the wrong cluster. The average silhouette score for a different number of clusters is represented in Figure 6.8. From this analysis we chose to split the transformed points into 3 clusters (Figure 6.9). The symmetry in the PCA is due to the permutational symmetry of the sites A and B, i.e. each structure appears twice as ABC_2D_6 and BAC_2D_6. The average attention weights and softmax weights within the 3 clusters are shown in Figure 6.18. The matrices representing the attention weights have as rows the element which representation will be updated, and as columns the elements that affect the site that will be updated. In cluster 1, all sites are mostly affected by the element at site C, while clusters 2 and 3 are dominated by sites B and A, respectively. The softmax weights show that the valence band of cluster 1 is dominated by by A and B equally and C in smaller measure, while in cluster 2 and 3 both the band edges are dominated by site B and A respectively. Note however that in this case the contribution of elements to the band edge does not necessarily reflect the physics behind the formation of bands, but rather the way our neural network places the energy levels to calculate a gap that matches the training gap.

Figure 6.11 shows the element distribution for each site in the 3 clusters. The distribution is normalized so that each row adds up to one. The maximum of the scale is reduced to 0.5 to improve the contrast. In Figure 6.12 the element distribution at each site is shown in the different clusters, normalized in a way that each column (i.e. each element) adds up to 1. In cluster 1 the attention weights give high importance to the element at site C, so in this case alkali earths, Tl, Pb, Nb and N. These structures

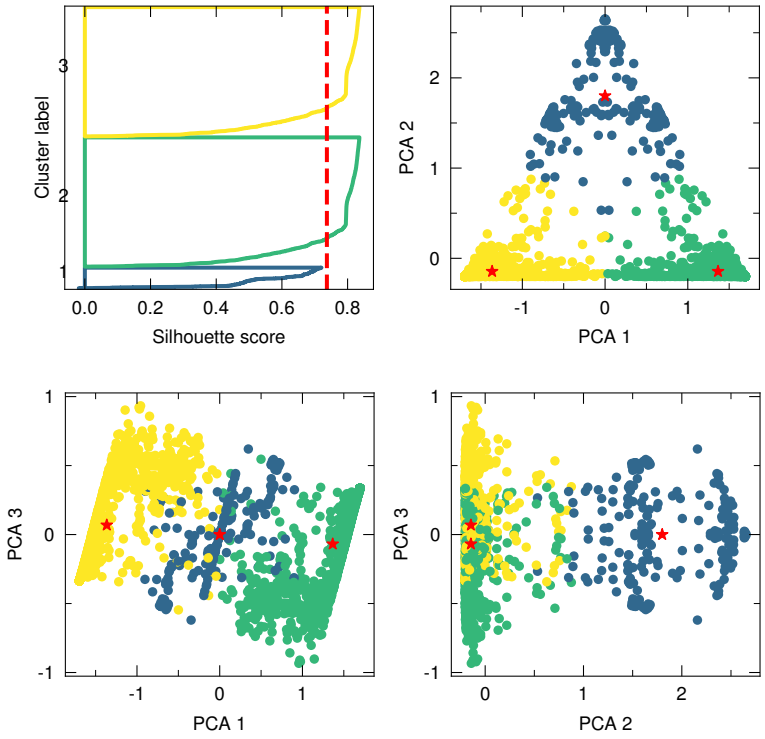
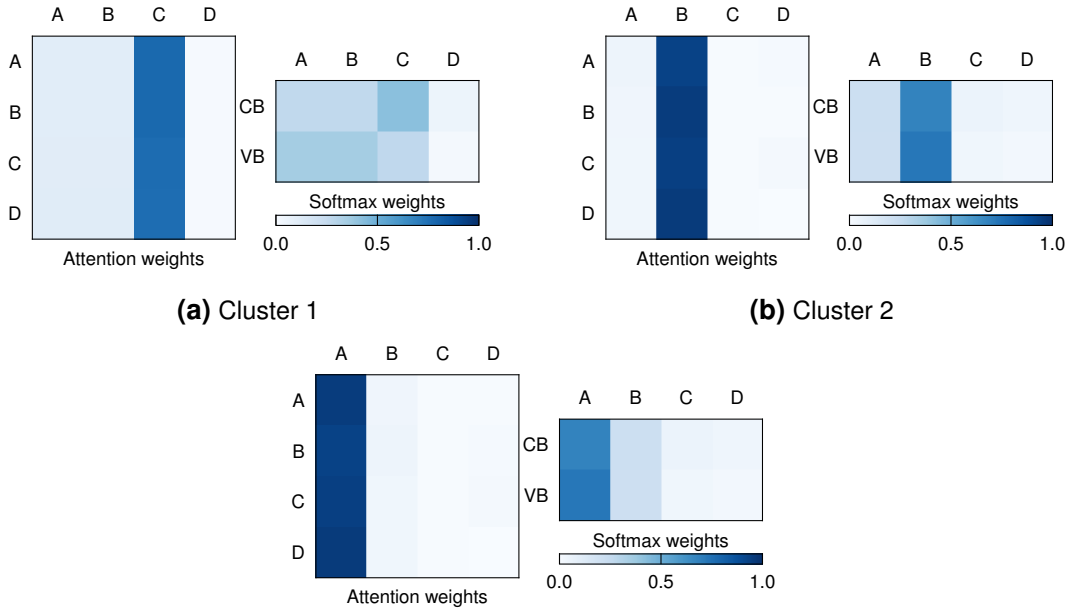


Figure 6.9 Silhouette scores and PCA for 3 clusters



(a) Cluster 1

(b) Cluster 2

(c) Cluster 3

Figure 6.10 Average attention weights and softmax weights in the 3 clusters. The rows in the attention weights represent the site where the element to be updated is, the columns represent the site at which the element that will be used for the update is.

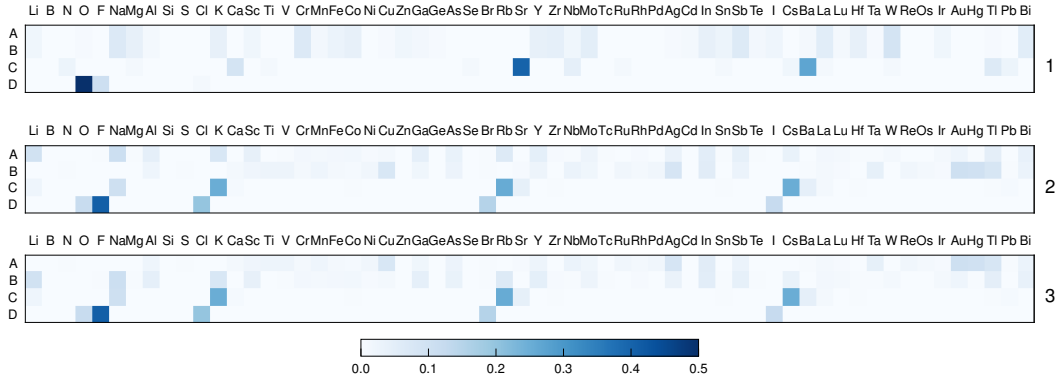


Figure 6.11 Normalized element distribution in the 3 clusters, with the color scale reduced to 0.5 to improve the contrast.

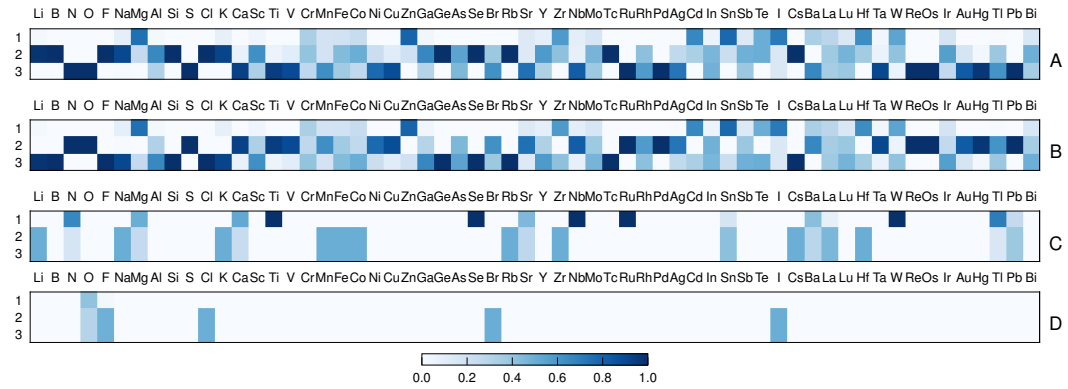


Figure 6.12 Normalized element distribution on the 4 sites

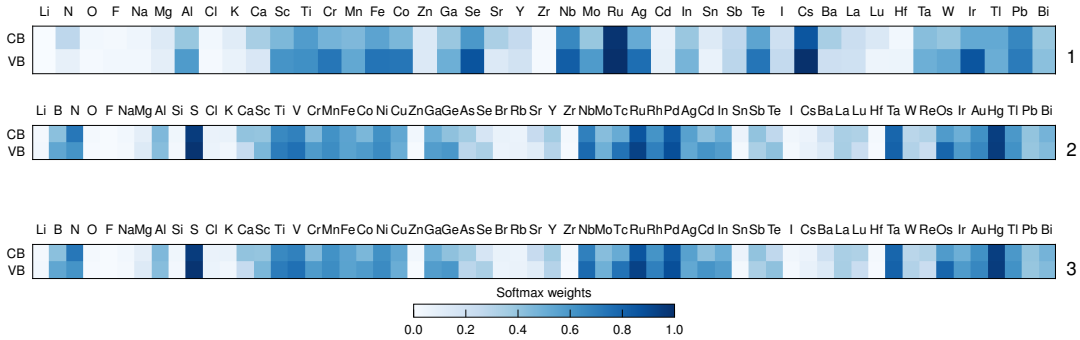


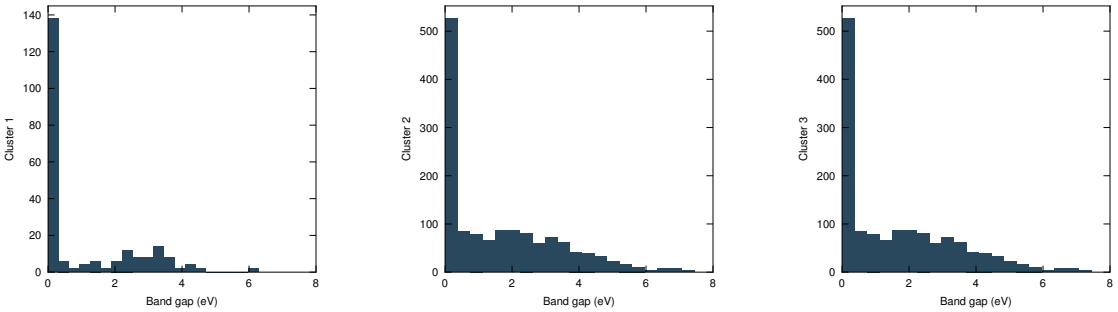
Figure 6.13 Average softmax weight by element in the 3 clusters. The elements shown are only those that appear in the cluster. A value of zero indicates that the element appears in the cluster but the softmax weight is zero.

are oxides for alkalin earths, Pb and Nb, and fluorides for Tl and N. From Figure 6.7 we can see that the model has placed these elements close to the zero, which means that it will give them high importance.

In Figure 6.13 the average softmax weights divided by element are shown. All the 3 clusters here behave very similarly, and most of the elements have a similar effect on the calculation of both band edges. From Figure 6.14 one can see that the band gap distribution in cluster 1 is centered around 3 eV, while the other clusters have a distribution that is very similar to that of the training set.

6.5 Training With Prior

After training with the prior we calculated again the optimal number of clusters via the silhouette score, which is 6 clusters this time (Figure 6.16). The 3-dimensional PCA has a total explained variance of 94% (60% 23% and 11% for the three dimensions), and the clustering is shown in Figure 6.17). Again, the symmetry of A and B site is



(a) Cluster 1 **(b)** Cluster 2 **(c)** Cluster 3

Figure 6.14 Band gaps distribution in the three clusters.

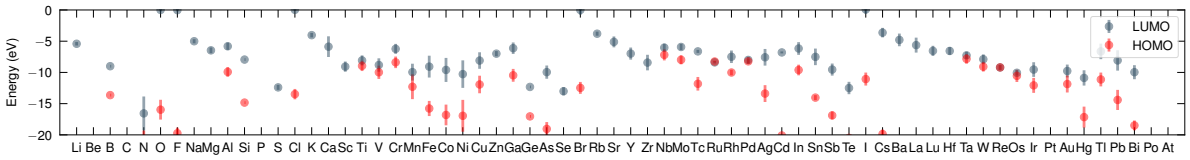


Figure 6.15 Predicted HOMO and LUMO values for a model trained with a prior ($\lambda = 0.001$), averaged over different structures, with standard deviation as error bar.

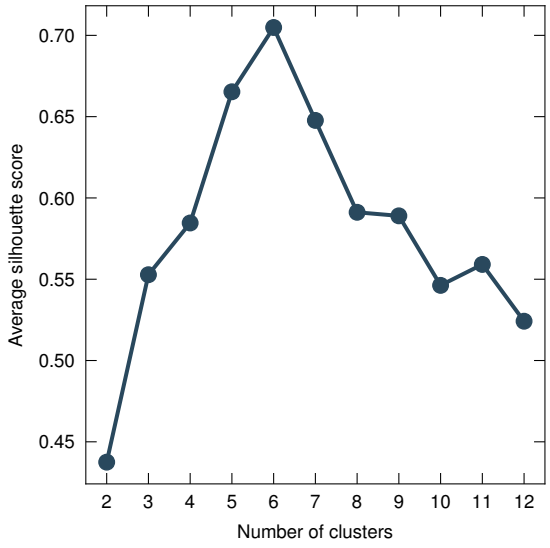


Figure 6.16 Average silhouette score for a different number of clusters

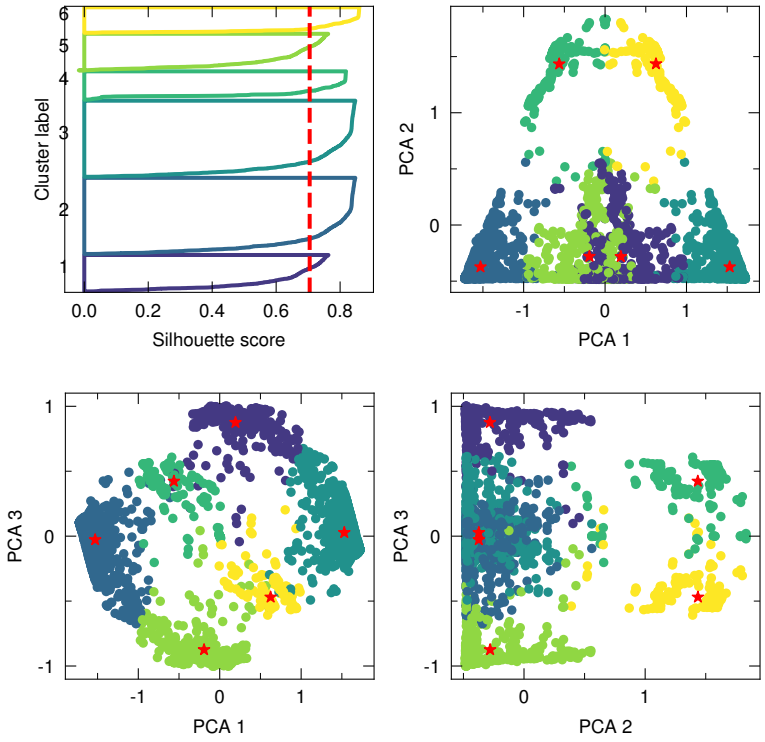
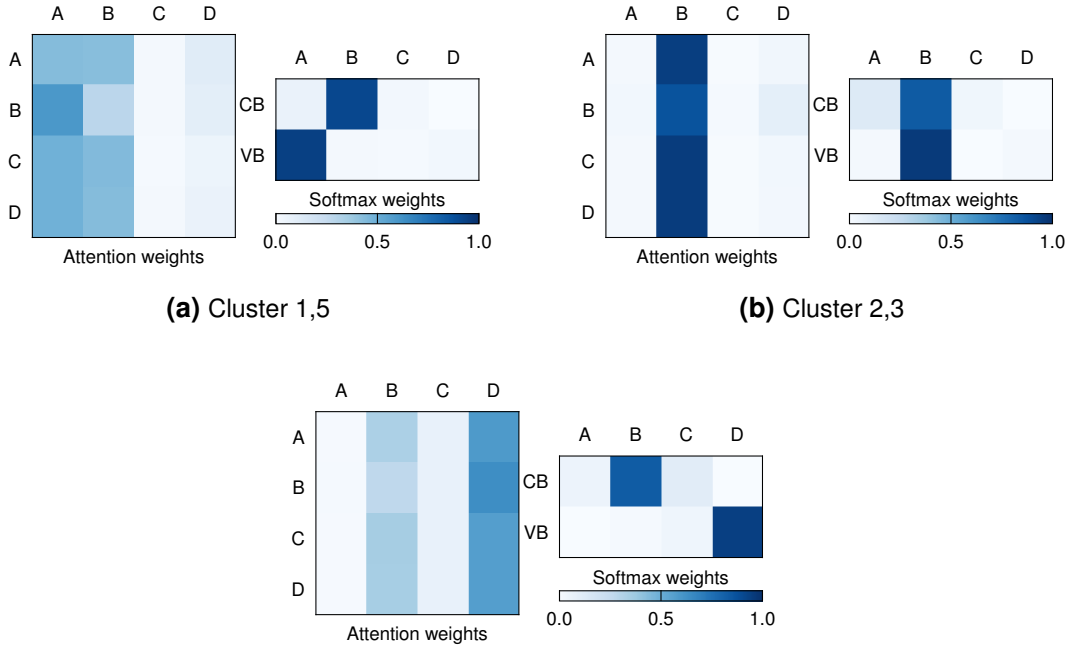


Figure 6.17 Silhouette scores and PCA for 6 clusters



(a) Cluster 1,5

(b) Cluster 2,3

(c) Cluster 4,6

Figure 6.18 Average attention weights and softmax weights in the 3 clusters. The rows in the attention weights represent the site where the element to be updated is, the columns represent the site at which the element that will be used for the update is. If A and B are permuted the clusters 1-5, 2-3, and 4-6 are equivalent.

reflected by the PCA, and specifically cluster 1-5, 2-3 and 4-6 are symmetric. From Figure 6.18 we can see that cluster 1 and 5 are dominated by site A and B equally (A for the VB and B for the CB in cluster 1 and the opposite for 5), cluster 2 and 3 by only B and only A, respectively, and cluster 4 and 6 by B and D (D for the VB and B for the CB), and A and D (again D at the VB and A at the CB). In the following we will show only the three independent clusters 1, 2 and 4. Cluster 5, 3 and 6 can be obtained by simply swapping A and B. Figure 6.19 shows the element distribution in each clusters, normalized in a way that the sum of each row (i.e. each site) is 1. This plot shows what elements are the most common in each cluster, and at what lattice site. Figure 6.20 shows the element distribution on the four lattice sites, normalized in a way that the sum of each column (i.e. each element) is 1. This highlights if an element on a lattice site appears more often in a specific cluster.

- Cluster 1 (5):
  - Site A (VB): Al, Cr, Cu, Ga, Nb, Mo, Ag, In, and Tl.
  - Site B (CB): Fe, Co, Ni, Cu, As, Ag, Sb, Hg, and Bi.

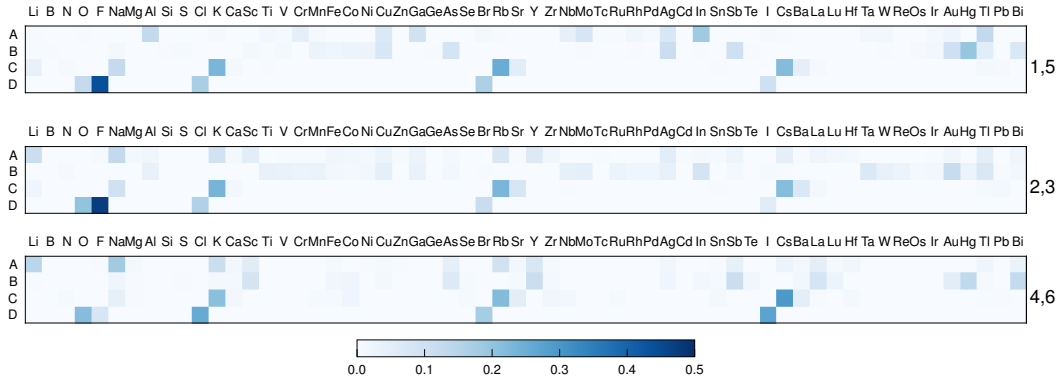


Figure 6.19 Normalized element distribution in the 3 independent clusters. The labels on the right indicate first the plot actually displayed (1,2,and 4) and second the cluster that is obtained by swapping the rows A and B (5,3, and 6) Each row in each subplot adds up to 1.

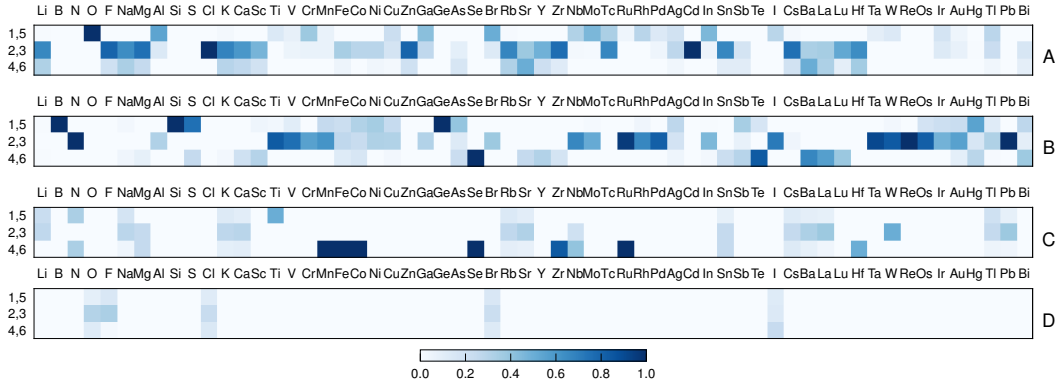


Figure 6.20 Normalized element distribution on the 4 sites. The labels on the left indicate first the plot actually displayed (1,2,and 4) and second the cluster that is obtained by swapping the rows A and B (5,3, and 6) Each column in each subplot adds up to 1.

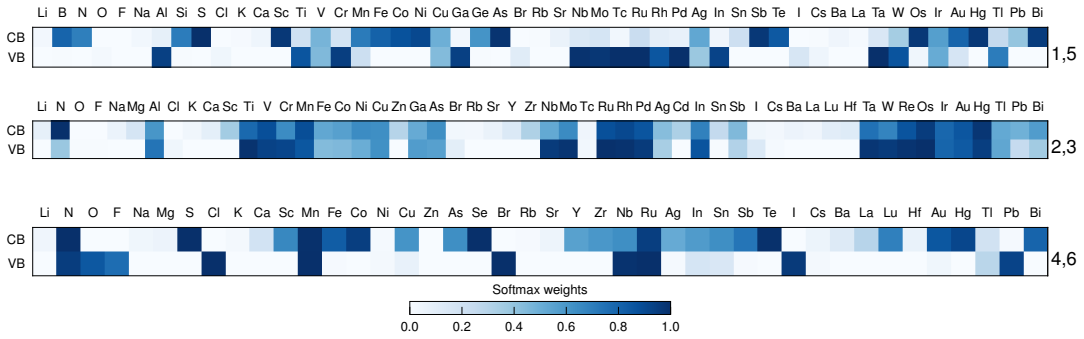


Figure 6.21 Average softmax weight by element in the 3 independent clusters

The VBM is calculated from elements with filled s or d states. The CBM is calculated from elements with empty d or p states. The band gaps of this clusters (Figure 6.22a) are distributed around 1 eV.

- Cluster 2 (3):
  - Site B (VB and CB): Al, Ti, V, Cr, Mn, Cu, Ga, Nb, Mo, Ru, Rh, Pd, In, Ta, W, Re, and Tl.

  Both VBM and CBM are calculated from transition metals and some elements of the p block. The gaps (Figure 6.22b) are distributed across a broad range and the majority of the metallic structures are found in this cluster. The HOMO and the LUMO of these metals (Figure 6.15)) are very close to each other.

- Cluster 4 (6):
  - Site D (VB): O, F, Cl, Br, I.
  - Site B (CB): K, Sc, Fe, Co, As, Y, Sb, La, Lu, Au, Hg, and Bi.

  The VBM is calculated from the anions (oxygen and haides) while the conduction band is calculated from metals of the p block, closed-shell transition metals or lanthanides and in a few cases from Fe and Co. Here the band gaps are distributed around 3 eV (Figure 6.22c).

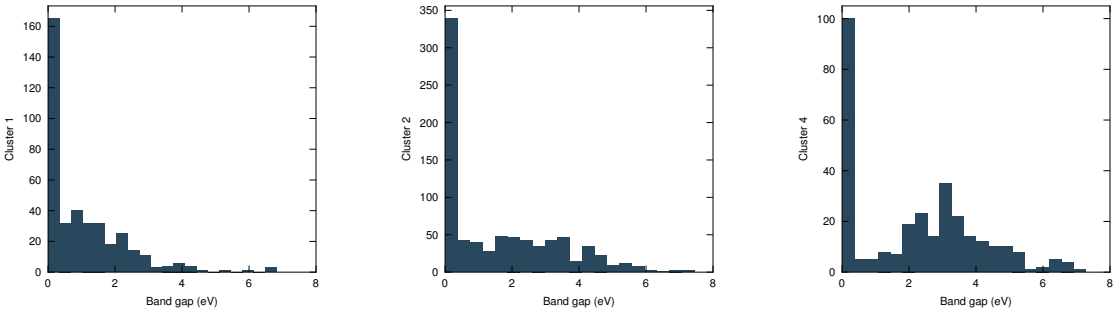
The elements at site C, mostly alkali metals and alkalin earths, have very low weight. If we look at the prior (Figure 6.4) which is reflected in the HOMO and LUMO after training (Figure 6.15) we can see that their energy levels are generally very deep in the bands, so the models will give them low weight. When they are at site A, they belong to clusters 2-3 and in smaller measure to cluster 4,6, while cluster 1-5 doesn't contain this kind of elements at site A-B (Figure 6.20).

Figure 6.21 represents the average softmax weight for each element in the 3 clusters, independent of their site. Also from here we can see that in cluster 1-5 and 4-6 different metals form the two are used to band edges while in 2-3 the elements have a similar effect on both edges.

Overall with this value of λ the model can distinguish more clusters with different properties that are heavily influenced by the prior, as can be seen from the small shift of the HOMO and LUMO values in Figure 6.15.

6.6 Training With a Weak Prior

If we repeat the same analysis with a lower constraint ($\lambda = 0.0001$) the model finds 4 clusters (Figure 6.24) with high weight on sites A, B, D, and C, with cluster 1 and 2 being symmetric (Figure 6.25). Figure 6.23 shows that the predicted HOMO and LUMO values still follow the prior but with a larger standard deviation between different structures. The elements at site A in cluster 1 that contribute to the calculation of



(a) Cluster 1,5 **(b)** Cluster 2,3 **(c)** Cluster 4,6

Figure 6.22 Band gaps distribution in clusters 1-5, 2-3, and 4-6

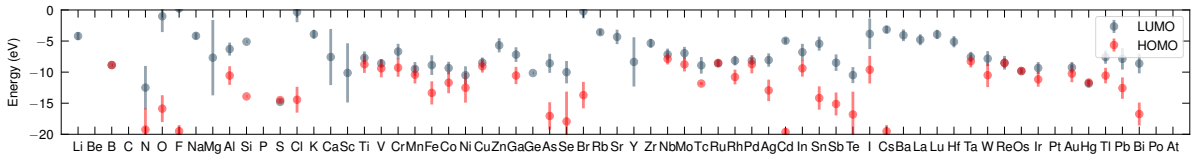


Figure 6.23 Predicted HOMO and LUMO values for a model trained with a weak prior ($\lambda = 0.0001$), averaged over different structures, with standard deviation as error bar.

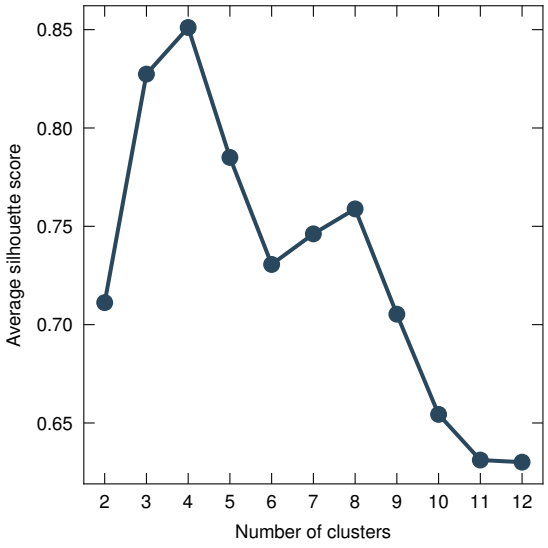
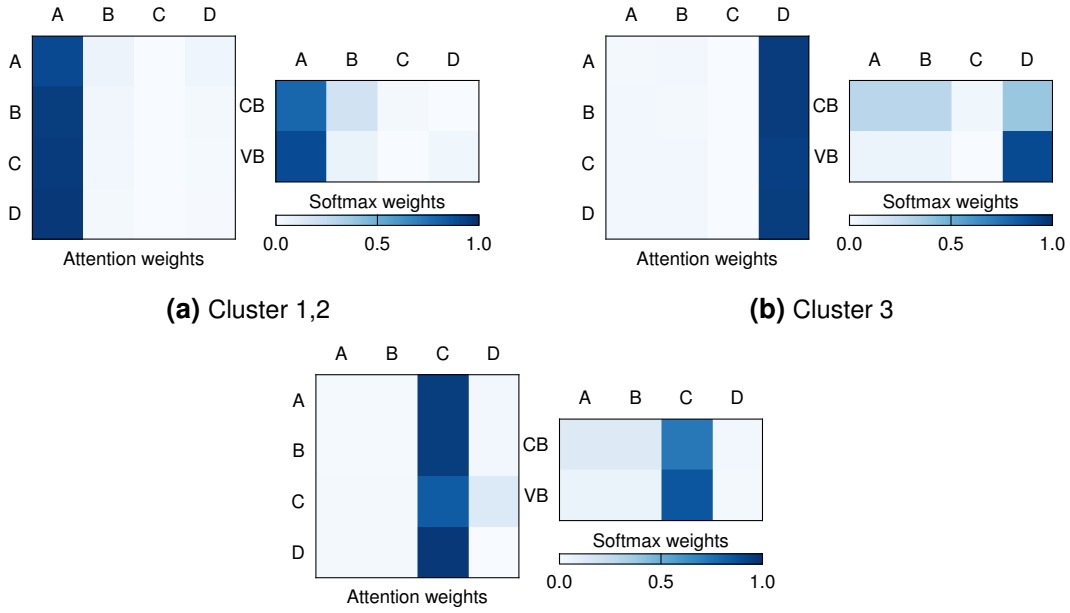


Figure 6.24 Average silhouette score for a different number of clusters



(a) Cluster 1,2

(b) Cluster 3

(c) Cluster 4

Figure 6.25 Average attention weights and softmax weights in the 4 clusters. The rows in the attention weights represent the site where the element to be updated is, the columns represent the site at which the element that will be used for the update is. If A and B are permuted, cluster 1 and 2 are equivalent.

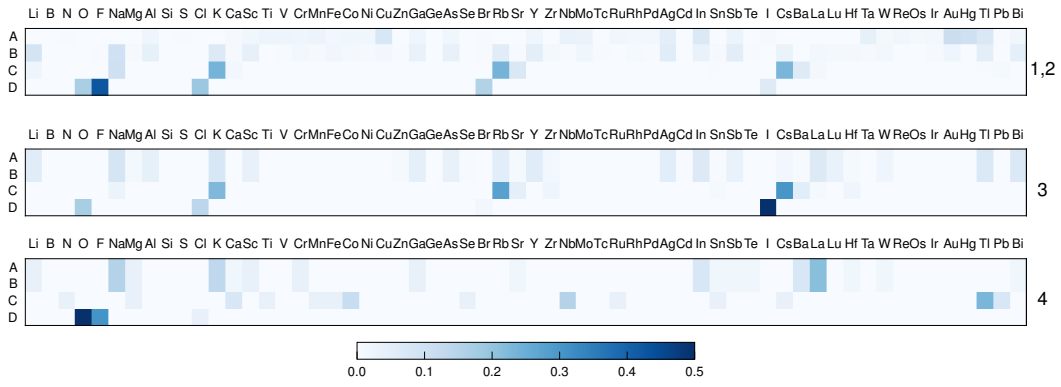


Figure 6.26 Normalized element distribution in the 3 independent clusters. Each row in each subplot adds up to 1.

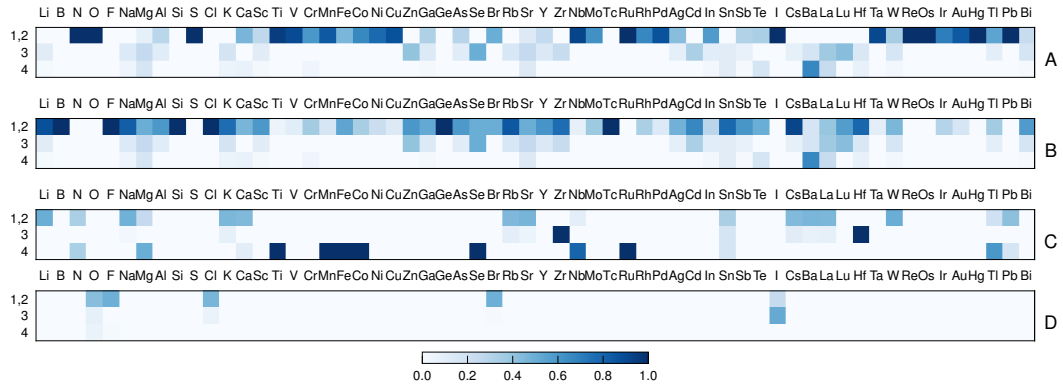


Figure 6.27 Normalized element distribution on the 4 sites. Each column in each subplot adds up to 1.

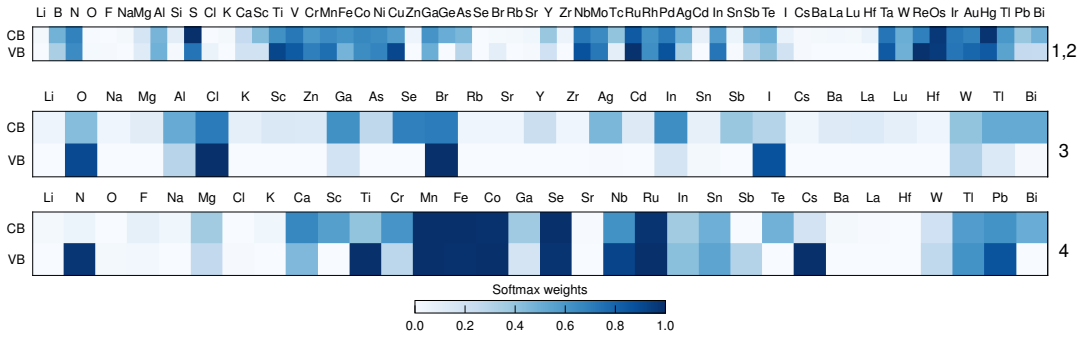
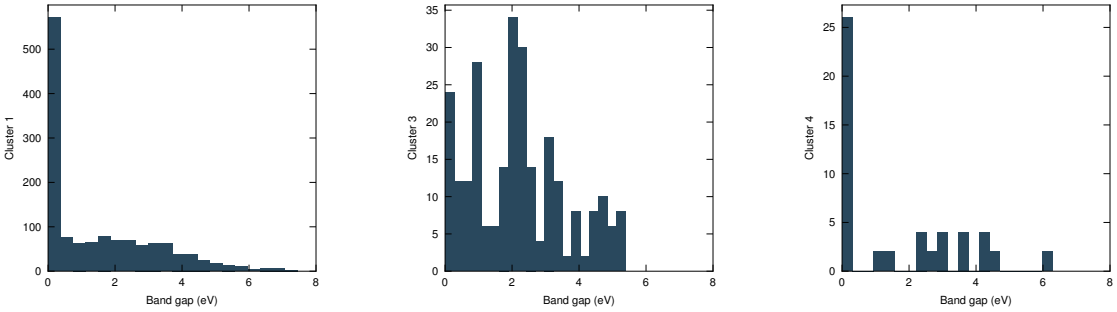


Figure 6.28 Average softmax weight by element in the 3 independent clusters



(a) Cluster 1,2 **(b)** Cluster 3 **(c)** Cluster 4

Figure 6.29 Band gaps distribution in clusters 1-2, 3, and 4

both valence and conduction band are mostly transition metals. Cluster 3, that has a valence band dominated by D and conduction band by A,B and D, contains O and halides at D and metals without unpaired electrons at site A-B. Cluster 4, dominated by C at both VB and CB, contains at C alkalin earths, N,Tl, Pb, and in a few cases Mn, Fe, Co, Se, Ru, Ti. At site A-B there are again elements without unpaired electrons. The band gap distribution (Figure 6.29) shows that cluster 1-2 is the largest one and resembles the most the distribution of the training set. Cluster 3 contains 129 unique structures (258 including permutations), with a gap centered around 2eV. Cluster 4 contains 26 unique structures (52 including permutations) and has band gaps divided in small clusters. An analysis of the content of this cluster shows that Nb, Mn, Co, Ru, N, Se, Fe and in one case Cs at site C form structures with zero gap, Sn and Ti have a gap of approximately 1 eV, Pb and Tl of 2 eV, Ca of 3 eV, Tl also 4 eV and Mg 6 eV.

6.7 Conclusions

In this work a message passing neural network was used to predict electronic energy levels of single atoms in the valence and conduction bands of double perovskites. The idea was inspired by the concept of Solid State Energy. A self-attention mechanism was used to update the representation of each element during the forward passes of the network, to highlight the pair-wise interactions between elements. The position of the band edges were calculated from the electronic energy levels as a weighted average with a softmax weight, and trained to fit the band gap. First we have shown that by initializing the energy levels as small random numbers or by initializing them based on a physics-inspired prior, the model is able to predict the gap with high and comparable accuracy. When the model is not provided with a prior, the predicted HOMO and LUMO values for each elements are symmetric respect to zero, and they contribute equally to the calculation of the position of the band edges. To analyze the weights of the network we defined a descriptor with the attention weights and softmax

weights and ran a PCA followed by k-means clustering. We found 2 clusters, one with high weight on site C, containing mostly oxides and with a band gap distribution centered around 3 eV, and another cluster with high weight on site A-B and a band gap distribution similar to that of the full training set.

Successively, we set an initial reference for the electronic energy levels with a penalty term of 0.001, and found 3 clusters with high weight on A and B, on only B, and on B and D. The first cluster contains metals at A-B, with the exception of alkali metals, that form the valence and conduction bands. To the second cluster belong most of the structures with two transition metals at A-B, that contribute equally to both band edges. This cluster has a band gap distribution that is very broad, and contains most of the metallic structures. The third cluster is dominated by the anions at site D that form the valence band, and by metals that form the conduction band, mostly without unpaired electrons.

We also performed the same analysis with a weaker prior, and found one cluster with most of the transition metals and a broad band gap distribution, dominated by site A-B, one cluster dominated by site D with metals without unpaired electrons at A-B forming the conduction band and anions at D forming the valence band, and one small cluster dominated by C with a few elements that are rarely found at this site.

From this analysis we can conclude that the presence of a prior and the penalty weight does not have a strong influence on the accuracy in the band gap prediction, but it heavily influences the way the model places the attention weights. In absence of a prior only two categories appear (oxides and others), while with a strong prior the attention follows the ordering of the energy levels defined by the prior. With a weak prior the models sees two broad categories (with and without transition metals) and one small group of outliers.

This results show that the attention weights and softmax scores can be used to gain some insights into how the materials can be clustered according to a target property. However, the clustering is not unique and depends on the details of the model, such as the strength of the prior in the loss function. While this approach thus offers some inspiration for potential composition/property relationships, these are not definitive and should be combined with rigorous physical analysis.

7 Assessing Deep Generative Models in Chemical Composition Space

7.1 Motivation

While the performance of predictive machine learning models for regression or classification can be easily assessed by calculating their accuracy on a test set, this is task is not trivial for generative models. In this case the model needs to reproduce the underlying data distribution and generate a high number of new and diverse samples, and in the case of conditional generation, that display a target property. Some benchmark studies have been done on organic molecules [192–195] and inorganic materials [196]. In this work [197] we focus on the generation of inorganic materials with a fixed structure and different chemical composition, with the formation energy as target property. We compare the performance of three deep generative models (VAE, GAN, and RL) trained on the well-established Elpasolite dataset by Faber et al. [115]. This dataset consists of a fully enumerated set of 2 millions structures of chemical formula ABC_2D_6 and their formation energy, and was already used in the past as a benchmark for regression models [198–202]. Using the full dataset as ground truth, we chose some performance metrics in regard to the precision of the models, the coverage of the target property class, and the fitness of the generated elemental distribution. These metrics, when defined only in relation to the training set, were also adopted to search for on optimal set of hyperparameters for the three models. The work summarized in this chapter has been published in Reference [197].

7.2 Dataset and Materials Representation

The chemical space considered in the original paper [115] consists of Elpasolites with chemical formula ABC_2D_6 (Figure 7.1a) with all the permutations of 83 elements (from H to Bi) in the 4 lattice sites. This structure prototype derives from the Elpasolite mineral $NaAlK_2F_6$, and it is equivalent to the double perovskite structure. In their paper, Faber et al. predicted the formation energy of these materials via Kernel Ridge Regression, obtaining a Mean Absolute Error (MAE) of 0.1 eV, for a total of 1,974,024 chemical formulas (Figure 7.1f). However, given that the lattice sites A and B are equivalent in the Elpasolite structure, i.e. that the chemical formulas ABC_2D_6 and BAC_2D_6 are actually the same compound, the number of unique structures that can

be formed with these 83 elements is halved. The original training set is consists of 10,590 structures with their DFT formation energy, including 34 compounds that appear as duplicates of the type ABC_2D_6/BAC_2D_6. Here we decided to first remove the duplicates and then double the size of the training set by adding the permuted form of each structure, with the same formation energy as the non-permuted one, to make sure that our models learn the intrinsic symmetry of the lattice sites A and B. The overall size of the training set, included permutations, is 21112 samples. The structure representation adopted in this work, inspired to the original paper by Faber et al., is an 8-dimensional vector x_{8D} with the shell number and group number of each element as entry. This vector has been successively scaled to have values included in the interval $(-1, 1)$, as this is beneficial for the training of neural networks.

$$x_{8D,scaled} = \frac{2 \cdot x_{8D} - 1}{n_{\text{row/group}}} - 1 \quad . \tag{7.1}$$

Note that this representation allows the generation of non-existent elements (for example an element between H and He), or structures that don't belong to our search space, such as structures with repeated elements. However, the generation of structures is not a time-limiting step, so we decided to simply discard these invalid structures when they occur during the generation process. In the following, we always refer to valid generated samples. To enable our models to generate materials conditioned on their formation energy, we divided the training set energies in 10 bins of equal size, and associated to each of them a class number from 1 for the lowest energy, to 10 for the highest energy (Figure 7.1e). Each energy class label has been represented as a 10-dimensional one-hot encoded vector.

7.3 Performance Metrics

What is required from these three generative models is the ability to generate structures in the correct energy class with high precision, while also generating as many new structures as possible, to cover the whole chemical space. These two requirements are not easy to satisfy at the same time, because a model that is very well conditioned on the energy could be able to generate only a few diverse structures, while a model that can generate many different structures might not necessarily be well conditioned. By exploiting the knowledge of the full Elpasolite space as calculated by KRR, the following metrics have been defined to quantitatively assess the performance of the three models.

1) *Precision (right class)* is the fraction of generated compositions that actually fall into the desired class.

$$\text{Precision (right class)} = \frac{N^{\text{gen}}_{\text{class}}}{N^{\text{gen}}} \quad : \tag{7.2}$$

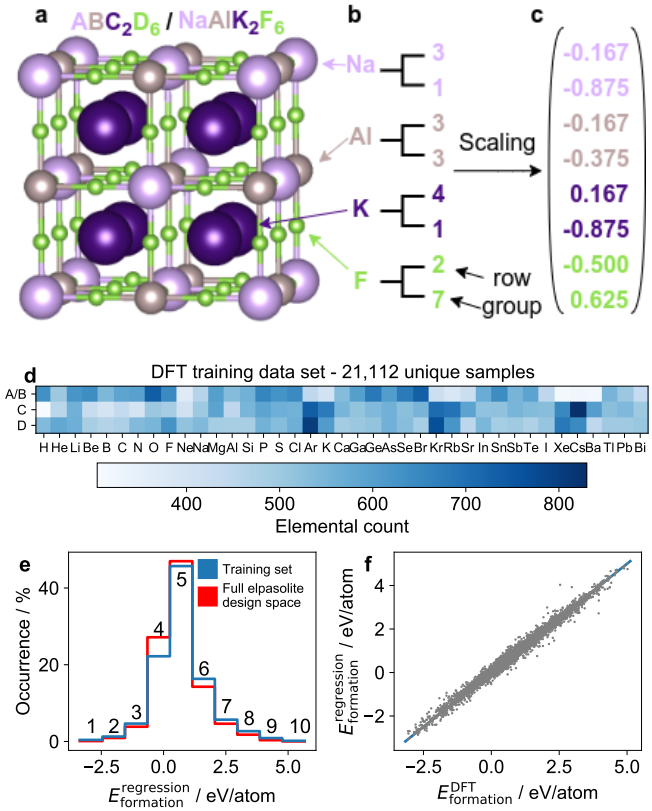


Figure 7.1 a) Elpasolite structure ABC_2D_6. b) 8D representation of the Elpasolite elemental composition by row and main group number in the periodic table, exemplified for $AlNaK_2F_6$. c) Scaling of the 8D representation by eq. (7.1). d) Elemental distribution of the training set. e) Formation energy distribution of DFT training set and the full reference dataset, split into ten energy classes. f) Formation energy of the DFT data vs. the predicted one from the regression model in the original paper. Figure reproduced from [197] under CC BY 4.0.

Here N^{gen} is the total number of generated samples and $N^{\text{gen}}_{\text{class}}$ is the number of generated samples which belong to the desired class.

2) *Precision (neighboring class)* is the fraction of samples falling into the classes just above and below the requested class.

$$\text{Precision (neighboring class)} = \frac{N^{\text{gen}}_{\text{class}-1} + N^{\text{gen}}_{\text{class}+1}}{N^{\text{gen}}} \quad . \tag{7.3}$$

This gives information about the tails of the predicted formation energy distributions.

3) *Coverage (right class)* measures the fraction of unique generated compositions in the desired class:

$$\text{Coverage (right class)} = \frac{N^{\text{gen}}_{\text{class,unique}}}{N_{\text{class}}} \quad . \tag{7.4}$$

4) *Coverage (neighboring class)* analogously measures the fraction of unique compositions in the classes just above and below the requested class,

$$\text{Coverage (neighboring class)} = \frac{N^{\text{gen}}_{\text{class}-1,\text{unique}} + N^{\text{gen}}_{\text{class}+1,\text{unique}}}{N_{\text{class}-1,\text{unique}} + N_{\text{class}+1,\text{unique}}} \quad . \tag{7.5}$$

5) *JS-Distance* measures how strongly the elemental distribution of the generated samples differs from the corresponding distribution in the full Elpasolite space, for a given class. Specifically, we use the Jensen-Shannon (JS) distance d_{JS} between the two elemental distributions p and q, defined as

$$\begin{aligned} d_{\text{JS}}(p||q) = \frac{1}{4}\sum_S \sum_Z \Bigg[& \frac{1}{2} p(Z|S) \log_2 \left(\frac{p(Z|S)}{m(Z|S)} \right) \\ & + \frac{1}{2} q(Z|S) \log_2 \left(\frac{q(Z|S)}{m(Z|S)} \right) \Bigg]^{\frac{1}{2}}, \end{aligned} \tag{7.6}$$

Here $p(Z|S)$ indicates the probability p of finding the element Z on the site S, and $m(Z|S) = \frac{1}{2}[p(Z|S)+q(Z|S)]$. The first sum runs over all sites A, B, C, D, the second sum runs over all elements in the Elpasolite dataset. This statistical distance takes values between 0 if the two distributions are identical and 1 if there isn't any overlap.

Figure 7.2 shows some reference values for the JS-Distance when we compare the elemental distributions of each class to the elemental distribution of every class in the full space. This shows that when the class is the same (e.g. class 5 in the training set compared to class 5 in the full chemical space), the JS-Distances is low, while different classes have high distances. Additionally, the distance between corresponding classes is lower when the class itself includes many samples (e.g. class 5), respect to classes with few samples (e.g. class 1) . Furthermore, the distances vary progressively across the classes, so the distance between class 1 and class 2 is lower than the distance between class 1 and class 5.

7.4 Hyperparameter Search

The architecture search was performed by randomly sampling 100 models from the domain shown in Table 7.1, and sorting them by performance. However, in a realistic situation the underlying data distribution (here the full Elpasolite space) is not known, so we chose the best performing models based solely on information that is available from the training set. To this end, we need to find a trade-off between two metrics:

1) JS-Distance between the generated distribution and the training set distribution of the corresponding class (here we chose to train the models on class 1). This value needs to be low to correctly reproduce the elements distribution, but it should not be zero, because that would lead to overfitting.

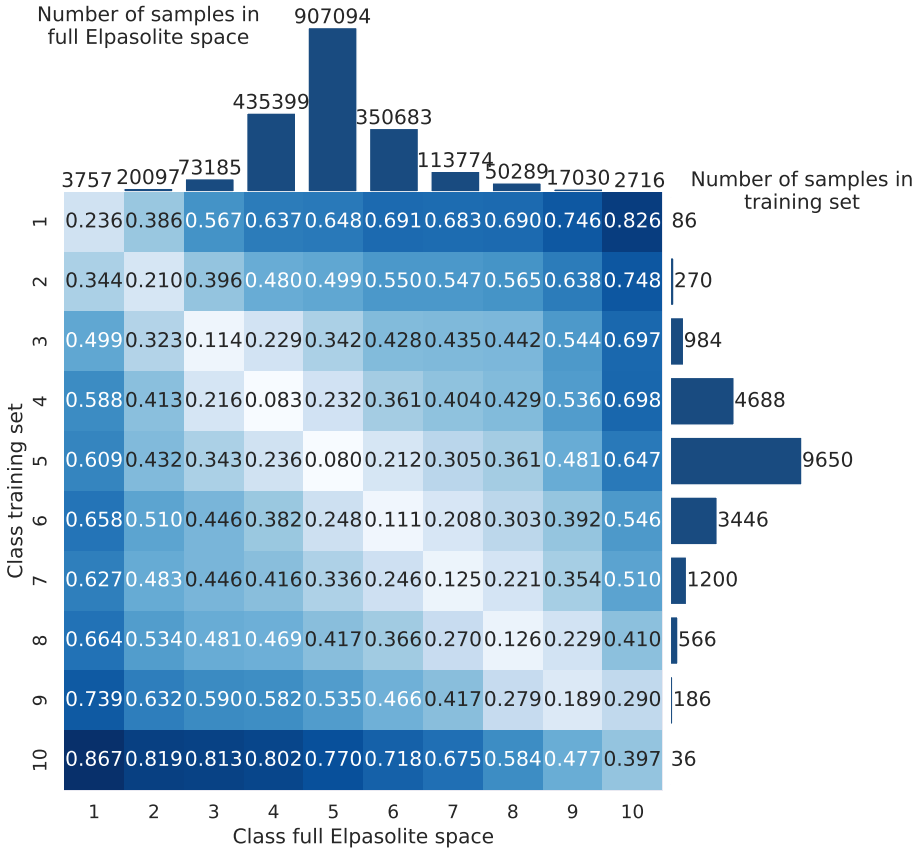


Figure 7.2 Reference *JS-Distances* between training set and the full Elpasolite space. Figure reproduced from [197] under CC BY 4.0.

2) Number of unique compositions generated. This value should be high, to make sure that the models are able to discover new structures outside the training set.

Figure 7.3 shows the results for the first 40 models sorted by JS-Distance. Due to the fact that different random initializations of the weights can produce slightly different results, we initialized each model 5 times and provided the median of the results. The performance metrics were calculated based on the generation of 10,000 samples from class 1. Full lines show the metrics that can be calculated from the training set, and were used to select a model, while dashed lines show other performance metrics that need the knowledge of the full space, for comparison. The optimal solutions lie on the left side of a pareto front (Figure 7.3, right). The vertical dashed lines show the selected models, i.e. those that show a good compromise between a high number of unique generate structures, and low statistical distance to the training set. The full results of the hyperparameter search are shown in Table 7.2, 7.3, and 7.4.

The selected models, highlighted in bold in the tables are:

Table 7.1 Search domain for random hyperparameter search. The listed hyperparameters influence the neural network architecture (top) and the iterative training process (bottom). Ranges reflect common values. Combinations of hyperparameters are drawn with replacement from the discrete sets of values. The hyperparameters are common to the VAE, GAN and RL model and the same combinations are used, with an increased number of training steps for RL due to the increased training set size. Note, that the learning rate is kept constant during training.

Parameter	Values
Number of layers	2, 3, 4, 5, 6, 7, 8
Nodes per layer	8, 32, 128, 256, 512, 1024
Latent space dimension**	2, 3, 4, 5, 6, 8
β***	5
Update steps*	1e4, 2.5e4, 5e4, 1e5, 1.5e5, 2e5
Batch size	10, 100, 250, 500, 1000, 2000
Learning rate	1e-6, 1e-5, 1e-4, 1e-3, 1e-2

*for RL, these values are multiplied by 8, **does not exist for RL, ***RL only

- VAE: the scaled input $x_{8D,scaled}$ is concatenated to the class one-hot encoded vector, then encoded to a 8-dimensional latent space, and decoded into the scaled 8-dimensional representation of the composition. Both the encoder and decoder networks have 2 hidden layers with 256 nodes. Layer normalization [165] is employed. The final output function of the decoder is a hyperbolic tangent that returns values between -1 and 1. The model is trained for 10,000 network updates on batches of 500 samples each, using the Adam optimizer [167] with a learning rate of 0.001. The regularization parameter λ in the loss is fixed at 0.1 [203].

- GAN: the scaled input $x_{8D,scaled}$ concatenated to the class one-hot encoded vector is passed to G_θ and decoded to the scaled structural representation, again employing a hyperbolic tangent output layer. The network is trained for 200,000 update steps on batches of 100 samples with a learning rate of 0.00001. The gradient penalty λ is 10.

- RL: the network architecture has five hidden layers with 512 nodes each, with layer normalization [165]. The RL model was trained for 800,000 network updates with a batch size of 250, and a learning rate of 0.00001. Following common practice in the DQN literature [204], a discount factor γ=0.999 is used and the target network is synchronized every ten steps. The reward was defined as

$$r(s,a) = \begin{cases} 0 & \text{if } s' \text{ non-final} \\ 1 & \text{if } s' \text{ final and in desired class} \\ -1 & \text{if } s' \text{ final and not in desired class} \end{cases} \tag{7.7}$$

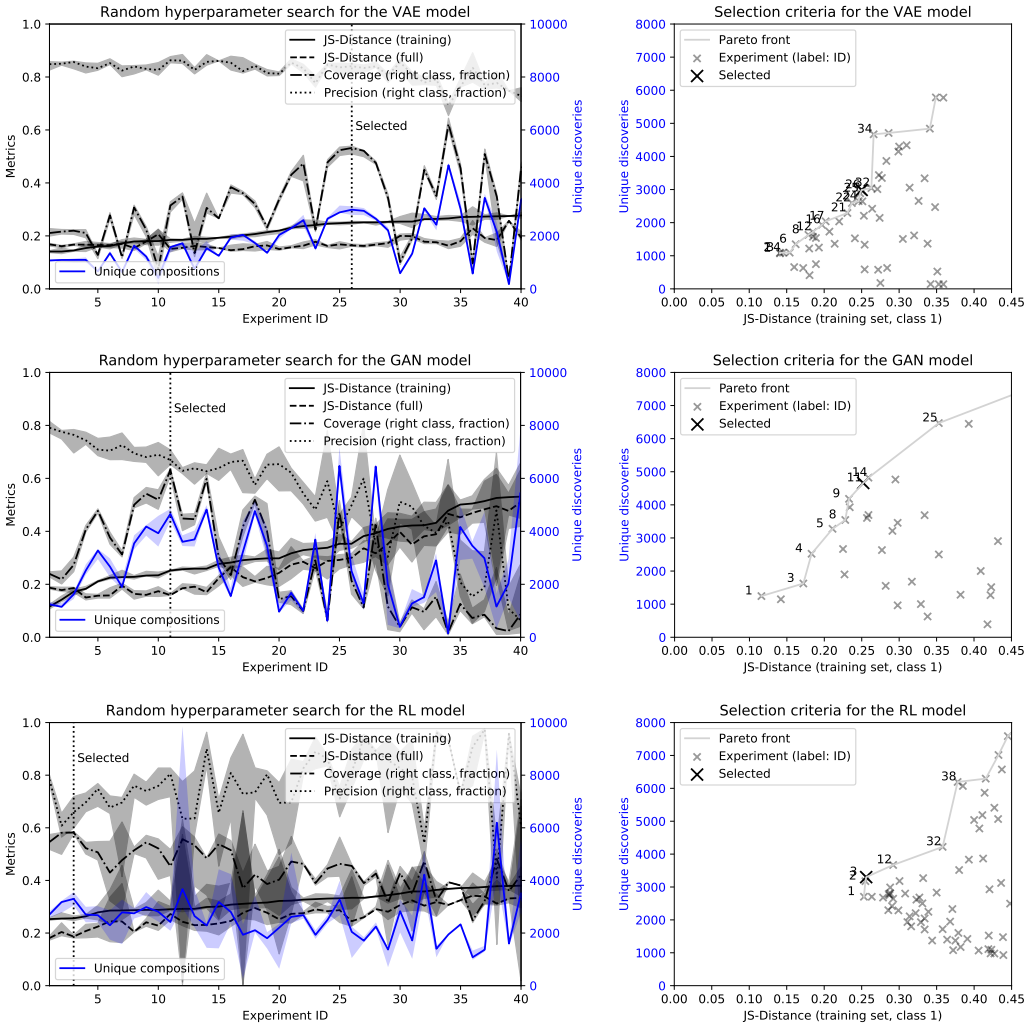


Figure 7.3 Results of the hyperparameters search for the first 40 models sorted in order of JS-Distance respect to the training set. The lines are the median value after 5 random initializations, the shaded area represent the minimum and maximum of the 5 values. Figure reproduced from [197] under CC BY 4.0.

7.5 Generation in a Minority Class

The first challenge for the generative models is to generate rare structures, for example in our case we will choose structures in the lowest energy class (class 1). This class is one of the the least populated and contains 86 samples, corresponding to 0.4% of the total training set. In the full Elpasolite space, class 1 contains 3,757 compositions, so these 86 samples are 2% of the whole class. Each model was trained 50 times with 50 different random initializations, and the results were averaged. After training the three models, we sampled 250,000 (valid) compositions as shown in Fig-

ure 7.4. The generated samples are classified as *novel discoveries* when they were

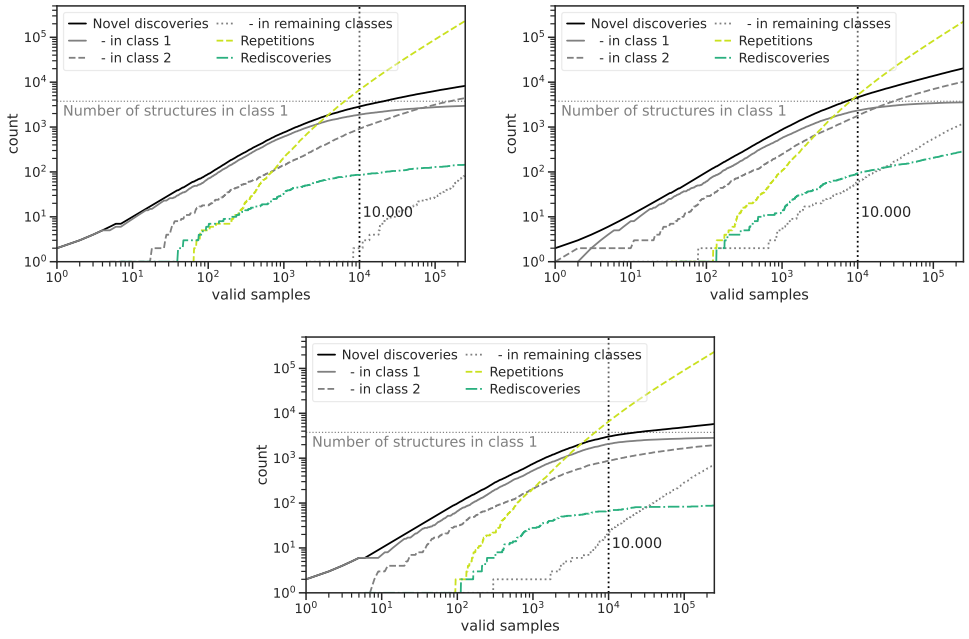


Figure 7.4 Performance of VAE (top left), GAN (top right) and RL (bottom) for the first 250,000 generated valid samples, when conditioned on the minority class 1. Figure reproduced from [197] under CC BY 4.0.

not previously found in the training set, *rediscoveries*, when they were already in the training set, and *repetitions* when they had already been generated by the model. The novel discoveries, were also divided in class 1 (the actual class that was requested), class 2 (the neighboring class), and remaining classes. As we can see from Figure 7.4, the models first generate many new structures, but then tend to saturate, as the repetitions prevail. The number of rediscoveries remains overall low. The novel discoveries fall mainly in the desired class, with some contribution from the neighboring class. This shows that the models are learning a class-conditional probability distribution, so at first the most likely samples appear, and then when repeating the sampling the probability of sampling from the tails of the distribution increases. Following these observations, we chose to sample 10,000 structures as threshold for saturation.

Figure 7.5 shows the energy distribution of the generated samples compared to the training set. As we can see, the three models generate a distribution peaked in the right class, with some tail in the neighboring one. This is likely a consequence of our choice to define classes with rigid boundaries, while the models produce more smooth distributions. From this we can see that the task of producing rare samples has been completed successfully. The metrics in Table 7.5 show the results quantitatively. All the three models show high precision, with more than 65% of the generated samples

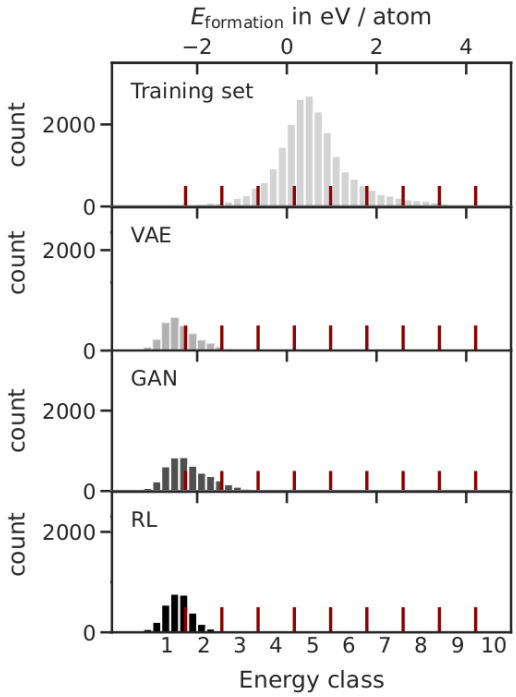


Figure 7.5 Formation energy distribution (in 0.2 eV/atom bins) of the unique class 1 conditioned compositions proposed by the three generative models (VAE, GAN, RL) over 10,000 valid samples. Additionally shown is the corresponding distribution over the entire training set of 21,112 compositions. Figure reproduced from [197] under CC BY 4.0.

falling in the right class, an overall more than 94% falling in class 1 and 2. The coverage is also above 50% for all the models. The low JS distance shows that the models are able to reproduce the correct element distribution, even better than the training set class 1, which has a JS distance of 0.236 to the full space class 1. Figure 7.6 shows the normalized element distribution over the 4 lattice sites. Some insight in how to generate Elpasolites with very negative formation energy can be gained from this. Class 1 shows some prominent features, such as a prevalence of fluorine on site D, likely due to its high electronegativity. Other common features are alkali metals and alkaline earths on site C.

7.6 Baseline Model

As baseline model for the generation of samples in a minority class we chose an enrichment-based model. This model enables us to verify whether the generative models are learning the distribution of the required class, or they are simply memorizing combinations of elements that are likely to be found in the required class, and reproposing these combinations during the generation phase.

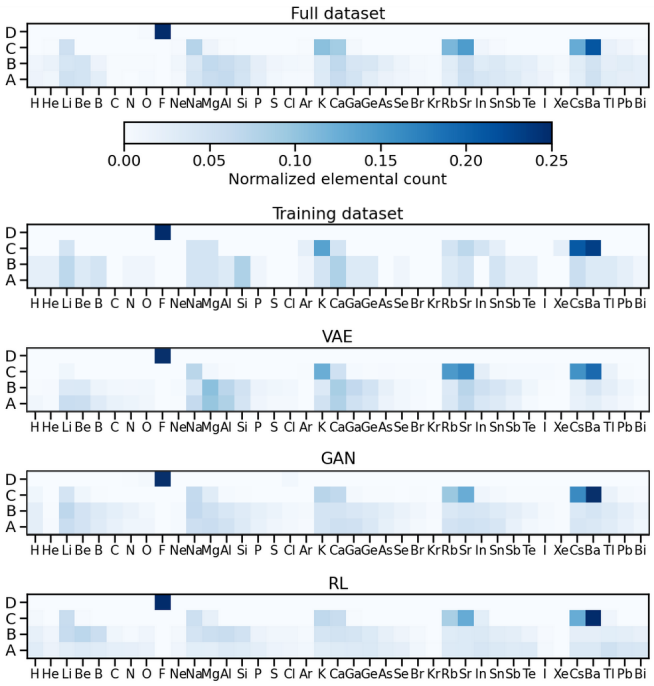


Figure 7.6 Normalized element distributions $p(Z|S)$ in class 1 over the four sites (A, B, C, D) for the full Elpasolite space, the training set and structures generated by the three models. The color scale is chosen to depict the details of the overall distribution. The value of fluorine on lattice site D surpasses its limits, exceeding 0.997 in all cases. Figure reproduced from [197] under CC BY 4.0.

To this end, we extracted from the training set all possible elemental triples of the form $\mathrm{NBC_2D_6}$, $\mathrm{ANC_2D_6}$, $\mathrm{ABN_2D_6}$, and $\mathrm{ABC_2N_6}$ where N is any other element from H to Bi. This was done separately for class 1 and for the remaining classes 2-10, yielding 344 triples in total and 320 unique ones. For each triple i in class 1, we calculated the enrichment score χ_i

$$\chi_i = \frac{(n^i_{\mathrm{class1}}/N^{\mathrm{structures}}_{\mathrm{class1}})}{(n^i_{\mathrm{class2-10}}/N^{\mathrm{structures}}_{\mathrm{class2-10}})} \tag{7.8}$$

where n^i_{class1} indicates the occurences of triple i in class 1 of the training set, $n^i_{\mathrm{class2-10}}$ the occurences of triple i in class 2-10, and $N^{\mathrm{structures_{class}}}$ indicates the total number of structures in the class ($N^{\mathrm{structures}}_{\mathrm{class1}}$ = 86 and $N^{\mathrm{structures}}_{\mathrm{class2-10}}$ = 21,026).

A high enrichment score of a triple then indicates a high probability of appearing in class 1. We then ranked the first 25, 50 and 100 triples based on this score, and used them to find class 1 structures in the full Elpasolite space. The results of this analysis are collected in Table 7.6. Among the best ranked unique triples, we found 2, 12 and 224 triples with the same enrichment score, each occurring 3, 2, and 1 times in class 1, and did not occur in classes 2-10 at all ($n^{\mathrm{i}}_{\mathrm{class2-10}}$=0), therefore we set $n^i_{\mathrm{class2-10}}$=0.1 to avoid division by 0. For the final selections of the 25, 50 and 100

best triples, we always include these 14 best ranked, while completing the selection by randomly drawing 11, 36 and 86 triples among the next best triples.

If we compare these values to the generative models (Table 7.5), we can see that enrichment scores allow the prediction of class 1 structures, but the generative models are significantly more accurate in terms of coverage, precision and JS-distance. This indicates that the generative models are learning higher order correlations that are not reflected in the enrichment scores.

7.7 Generation in the Majority Class

The second task for the generative models is to test their capacity, i.e. the ability to generate many distinct samples. This will be tested on class 5, which contains 907,094 elements and represents 46% of the full Elpasolite space. The training set contains 9,650 class 5 structures (46% of the training set), which are only 1% of the all the possible class 5 structures, so even a smaller fraction of the structures in class 1. The same VAE and GAN could be used to generate structures in class 5, while the RL model needs to be retrained on each separate class. We generated 25 million

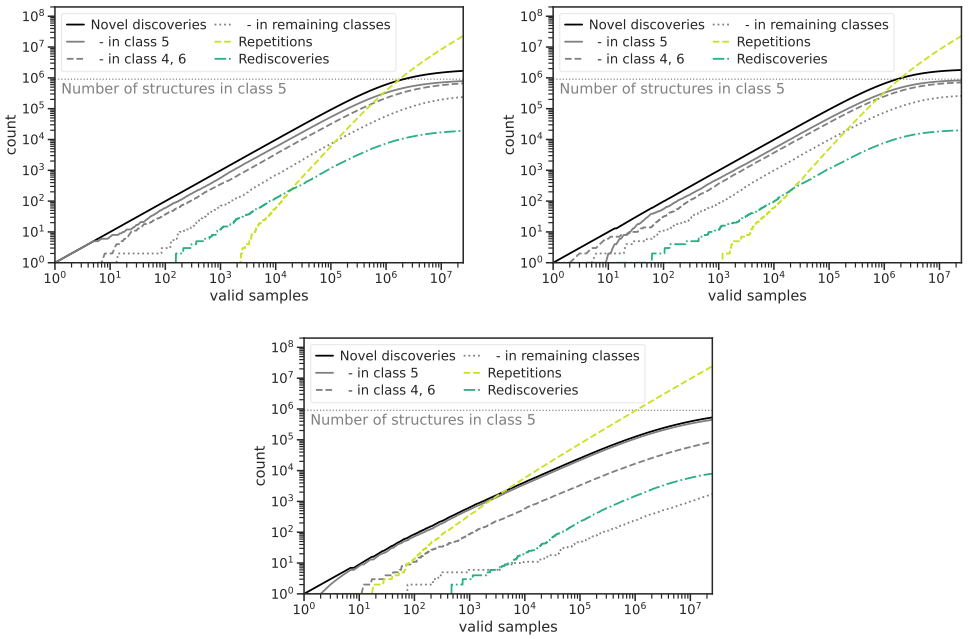


Figure 7.7 Same as Figure 7.4, but for a VAE (top left), GAN (top right) and RL (bottom) conditioned on the majority class 5. Figure reproduced from [197] under CC BY 4.0.

valid samples to make sure that the models are saturated (Figure 7.7) and calculated the same performance metrics.

Table 7.7 shows the calculated metrics. Both GAN and VAE achieve average class coverage of 86% or higher, with precision between 50 and 60%. In contrast, the RL model achieves significantly higher precision (87% on average), but lower coverage. Such trade-off between precision and coverage is not necessarily intrinsic to the models themselves, but it was present also during the hyperparameter search when comparing different architectures for the same model. For example, the RL could achieve higher coverage at the expense of lower precision by decreasing the inverse temperature β in Eq. 4.11.

However, the JS distance for the RL models is systematically larger than VAE and GAN. As we can see from Figure 7.8, the VAE and GAN produce an even element distribution across the periodic table, while RL focuses on a subset of elements. These subsets correspond to different local minima that can be reached by re-initializing the model. For this reason, the VAE and GAN, as distributional learners, seem to be more suited for this task.

7.8 Influence of Training Data

The third task for the generative models is to try to reduce the amount of DFT calculations they rely on. Up to now, the models were trained on a set of 21,112 structures that required 10,556 calculations due to permutational symmetry at the A/B site. This amounts to 1% of the full space. However, we would need to perform other calculations on the generated structures to verify that they actually belong to the right class. This amounts to a few thousands more calculations, which is still preferable to random sampling or brute-force screening, but is still a limitation. For this reason we reduced the size of the training set as shown in Figure 7.9. We also compared our results with models trained on the original training set without permutational symmetry. The results are shown in Figure 7.9.

This shows that models trained on 10,000 data points including permutations (which would require only 5,000 DFT calculations) reach the same performance as models trained on the original set, while reducing the number of DFT calculations by a factor of two. In general, we find that both precision and coverage for the target class decrease when the training set size decreases, and the JS-Distance between generated and target distribution increases. Here, the RL models seem less sensitive, retaining higher coverages for the smallest training sets.

However, even small training sets of 1000 samples yield models capable of discovering a significant number of new materials. Taking the GAN as an example, with training set size of 1000 compositions (500 DFT calculations), 11% ($>$ 410 samples) of all possible class 1 compositions are discovered, easily surpassing the 86 samples found in the original training set. With a precision of 50%, this means that approximately 1000 additional DFT calculations would be needed to obtain the formation energies of all generated samples. With random sampling, finding the same number of class 1 materials would require 220,000 DFT calculations (*i.e.* a factor 100 more).

It is also important to note here that all model hyperparameters were kept fixed, even if small datasets might need a different network architecture. Furthermore, random sampling is a fairly strong baseline in this example due to the limited size of the Elpasolite composition space. In reality, chemical space is practically unlimited and brute-force random search would not be a viable strategy.

7.9 Conclusions

We herein assessed the performance of three deep generative models (VAE, GAN and RL) for the exploration of a large comoposition space. The fully enumerated Elpasolite dataset enabled us to define metrics for precision, coverage and elemental distribution. Three simple model architectures were selected based on metrics that can be calculated solely from the training set. All studied models were capable of reliably generating candidates within the desired formation energy classes, with high precision and high coverage of the space. The RL models showed greater robustness when trained on small datasets, but also greater variance between differently initialized models (*i.e.* a tendency to converge to different local minima), while the VAE and GAN produced models that more reproduced the target elemental distributions, with lower the JS-Distance.

From a technical perspective, the RL models were retrained with a different reward function for each requested class independently, while the VAE and GAN can learn the distribution of all the classes simultaneously. Among these two, the VAE showed higher precision and was overall best suited for large scale materials discovery. However, if the objective was to simply to generate a limited number of high-quality samples, a target-oriented RL model may be the better choice. RL furthermore has the advantage that the reward function can easily be modified to accommodate more complex design targets.

Compared to regression models, which are robust and simpler to train, generative models might find better application in the discovery of exceptional materials that are underrepresented in the training set, such as in our minority class generation example.

VAE										
Experiment ID	1	2	3	4	5	6	7	8	9	10
JS-Distance (training set, class 1)	0.141	0.141	0.145	0.154	0.160	0.162	0.172	0.179	0.180	0.180
Unique compositions	1073	1092	1093	1094	658	1348	629	1631	1218	412
Coverage (right class)	21	22	22	21	13	28	12	31	22	8
Coverage (neigh. class)	2	2	2	2	1	2	1	3	2	1
Precision (right class)	85	85	86	84	83	85	82	84	83	83
Precision (neigh. class)	15	15	14	16	15	14	17	15	16	17
JS-Distance (full set, class 1)	0.168	0.161	0.167	0.167	0.165	0.165	0.170	0.154	0.179	0.207
Unique rediscoveries	89	90	90	89	80	93	76	94	86	62
Learning rate	1e-3	1e-4	1e-3	1e-3	1e-5	1e-4	1e-5	1e-4	1e-3	1e-2
Batch size	500	2000	1000	500	500	100	1000	2000	100	2000
Latent space dimension	3	3	3	5	2	4	2	4	3	2
Nodes per layer	256	256	128	256	512	1024	512	1024	256	512
Number of layers	6	6	7	8	6	8	7	8	6	5
Update steps	1.5e5	2e5	1.5e5	5e4	1e5	2e5	2.5e4	1e4	1e5	1.5e5
Experiment ID	11	12	13	14	15	16	17	18	19	20
JS-Distance (training set, class 1)	0.185	0.185	0.189	0.189	0.193	0.197	0.202	0.207	0.214	0.220
Unique compositions	1575	1720	747	1537	1248	1934	2040	1729	1362	2037
Coverage (right class)	31	35	13	31	27	38	36	32	23	34
Coverage (neigh. class)	2	2	1	2	2	3	3	3	2	4
Precision (right class)	86	86	83	86	86	85	82	84	81	81
Precision (neigh. class)	13	13	16	14	14	14	17	15	17	18
JS-Distance (full set, class 1)	0.151	0.155	0.162	0.158	0.154	0.146	0.150	0.161	0.164	0.150
Unique rediscoveries	87	92	73	92	90	94	97	91	79	97
Learning rate	1e-3	1e-5	1e-3	1e-3	1e-2	1e-5	1e-5	1e-4	1e-3	1e-6
Batch size	1000	250	2000	1000	500	100	1000	1000	500	1000
Latent space dimension	8	6	3	5	8	6	5	4	3	4
Nodes per layer	128	1024	32	128	1024	1024	1024	512	128	1024
Number of layers	8	8	8	7	4	8	5	7	4	5
Update steps	2.5e4	2e5	2e5	5e4	1.5e5	2e5	1e5	1e4	2.5e4	1e5
Experiment ID	21	22	23	24	25	26	27	28	29	30
JS-Distance (tr. set, class 1)	0.231	0.237	0.241	0.246	0.248	**0.250**	0.251	0.251	0.253	0.254
Unique compositions	2292	2593	1527	2651	2888	**2989**	2946	2644	2209	588
Coverage (right class)	43	47	22	48	52	**53**	52	48	34	10
Coverage (neigh. class)	4	5	3	4	5	**5**	5	4	4	1
Precision (right class)	85	83	78	85	84	**84**	83	84	81	85
Precision (neigh. class)	14	16	19	14	16	**15**	16	15	18	13
JS-Distance (full set, class 1)	0.157	0.163	0.178	0.162	0.168	**0.164**	0.162	0.167	0.173	0.200
Unique rediscoveries	94	92	77	95	89	**91**	96	100	81	41
Learning rate	1e-5	1e-3	1e-6	1e-5	1e-4	**1e-3**	1e-6	1e-6	1e-4	1e-3
Batch size	250	100	2000	500	250	**500**	500	100	250	1000
Latent space dimension	6	6	3	6	8	**8**	8	8	4	2
Nodes per layer	512	256	256	128	1024	**256**	512	1024	128	256
Number of layers	6	6	8	6	5	**3**	8	8	5	3
Update steps	1e5	2.5e4	5e4	2e5	2.5e4	**1e4**	1.5e5	2e5	2.5e4	1e4
Experiment ID	31	32	33	34	35	36	37	38	39	40
JS-Distance (tr. set, class 1)	0.254	0.263	0.264	0.266	0.271	0.272	0.273	0.274	0.275	0.277
Unique compositions	1334	3044	2420	4669	3013	578	3437	2144	177	3350
Coverage (right class)	19	45	35	62	46	10	51	35	3	45
Coverage (neigh. class)	3	6	5	10	6	1	7	4	0	7
Precision (right class)	81	78	79	69	76	83	77	78	75	73
Precision (neigh. class)	17	20	19	27	21	16	21	19	25	24
JS-Distance (full set, class 1)	0.199	0.178	0.176	0.163	0.180	0.229	0.192	0.184	0.257	0.194
Unique rediscoveries	56	91	64	82	77	31	89	51	22	90
Learning rate	1e-4	1e-5	1e-4	1e-2	1e-5	1e-5	1e-6	1e-5	1e-2	1e-6
Batch size	100	500	100	1000	10	10	500	1000	2000	10
Latent space dimension	3	5	4	8	5	2	6	6	3	5
Nodes per layer	512	256	256	32	1024	1024	1024	32	1024	1024
Number of layers	8	6	2	2	7	8	4	8	6	7
Update steps	1e4	2.5e4	1.5e5	1e4	5e4	1e5	5e4	2e5	1e5	1e5

Table 7.2 Hyperparameter search for the VAE. Chosen model highlighted in bold.

GAN										
Experiment ID	1	2	3	4	5	6	7	8	9	10
JS-Distance (training set, class 1)	0.116	0.142	0.172	0.183	0.211	0.225	0.227	0.228	0.233	0.234
Unique compositions	1244	1148	1624	2522	3276	2666	1898	3543	4181	3928
Coverage (right class)	24	22	27	41	48	38	31	50	54	52
Coverage (neigh. class)	2	2	3	5	7	6	4	7	9	9
Precision (right class)	79	78	76	74	71	70	73	69	68	69
Precision (neigh. class)	20	21	23	24	27	26	26	27	28	27
JS-Distance (full set, class 1)	0.187	0.178	0.186	0.150	0.157	0.158	0.195	0.164	0.160	0.174
Unique rediscoveries	92	94	92	102	101	74	66	104	90	94
Learning rate	1e-4	1e-5	1e-5	1e-5	1e-5	1e-4	1e-4	1e-4	1e-5	1e-3
Batch size	2000	250	100	1000	250	1000	2000	250	500	500
Latent space dimension	3	6	6	5	6	4	4	8	6	8
Nodes per layer	256	1024	1024	1024	512	512	1024	1024	128	256
Number of layers	6	8	8	5	6	7	8	3	6	3
Update steps	2e5	2e5	2e5	1e5	1e5	1e4	1e4	2.5e4	2e5	1e4
Experiment ID	11	12	13	14	15	16	17	18	19	20
JS-Distance (training set, class 1)	**0.252**	0.257	0.259	0.259	0.277	0.282	0.291	0.295	0.298	0.298
Unique compositions	**4660**	3604	3693	4821	2634	1553	3213	4766	3457	965
Coverage (right class)	**63**	45	45	60	30	23	42	52	40	14
Coverage (neigh. class)	**10**	8	8	10	6	3	7	10	7	2
Precision (right class)	**67**	63	64	63	64	66	67	57	65	65
Precision (neigh. class)	**28**	31	30	30	29	30	28	31	28	28
JS-Distance (full set, class 1)	**0.159**	0.185	0.191	0.170	0.223	0.265	0.227	0.211	0.224	0.243
Unique rediscoveries	**98**	77	76	97	59	53	79	96	75	35
Learning rate	**1e-5**	1e-6	1e-4	1e-6	1e-4	1e-4	1e-5	1e-5	1e-6	1e-5
Batch size	**100**	1000	250	500	100	100	10	500	100	500
Latent space dimension	**8**	4	4	6	4	4	5	5	8	2
Nodes per layer	**512**	1024	128	1024	256	1024	1024	256	1024	512
Number of layers	**2**	5	5	4	2	8	7	6	8	6
Update steps	**2e5**	1e5	2.5e4	5e4	1.5e5	2e5	5e4	2.5e4	2e5	1e5
Experiment ID	21	22	23	24	25	26	27	28	29	30
JS-Distance (training set, class 1)	0.317	0.329	0.334	0.338	0.353	0.353	0.382	0.393	0.409	0.418
Unique compositions	1679	1002	3688	628	6470	2502	1284	6447	2001	391
Coverage (right class)	16	10	29	7	47	21	11	42	12	4
Coverage (neigh. class)	3	2	7	1	12	5	2	8	5	1
Precision (right class)	62	54	48	59	41	50	60	35	30	51
Precision (neigh. class)	30	36	36	36	32	32	28	24	48	27
JS-Distance (full set, class 1)	0.274	0.285	0.261	0.288	0.292	0.300	0.336	0.317	0.352	0.398
Unique rediscoveries	34	20	65	13	101	45	18	94	32	8
Learning rate	1e-3	1e-5	1e-4	1e-3	1e-5	1e-6	1e-4	1e-6	1e-3	1e-5
Batch size	500	1000	2000	1000	10	500	100	2000	100	10
Latent space dimension	3	2	4	2	8	8	3	6	6	2
Nodes per layer	128	512	32	256	128	512	512	512	256	1024
Number of layers	4	7	2	3	3	8	8	5	6	8
Update steps	2.5e4	2.5e4	1e5	1e4	1e5	1.5e5	1e4	2.5e4	2.5e4	1e5
Experiment	31	32	33	34	35	36	37	38	39	40
JS-Distance (training set, class 1)	0.422	0.423	0.432	0.478	0.501	0.505	0.514	0.526	0.529	0.531
Unique compositions	1273	1514	2904	147	4167	3457	2989	1159	2030	5463
Coverage (right class)	11	10	15	2	13	7	9	3	2	9
Coverage (neigh. class)	2	2	4	0	4	3	4	1	1	3
Precision (right class)	49	42	39	44	21	15	19	48	11	6
Precision (neigh. class)	27	32	28	50	20	20	28	17	9	19
JS-Distance (full set, class 1)	0.350	0.381	0.389	0.466	0.453	0.469	0.483	0.495	0.477	0.510
Unique rediscoveries	24	21	44	4	64	44	40	14	22	87
Learning rate	1e-4	1e-4	1e-6	1e-3	1e-5	1e-6	1e-3	1e-3	1e-6	1e-4
Batch size	10	10	10	500	10	250	10	2000	250	1000
Latent space dimension	3	3	5	3	5	4	6	3	6	8
Nodes per layer	32	128	1024	256	256	256	128	32	128	8
Number of layers	4	6	7	6	5	4	5	8	7	2
Update steps	1.5e5	2.5e4	1e5	1.5e5	2.5e4	1e5	1e4	2e5	2e5	1e5

Table 7.3 Hyperparameter search for the GAN. Chosen model highlighted in bold.

				RL						
Experiment ID	1	2	3	4	5	6	7	8	9	10
JS-Distance (training set, class 1)	0.244	**0.249**	0.254	0.259	0.261	0.267	0.270	0.270	0.270	0.279
Unique compositions	2376	**2893**	2635	2579	2445	3177	2681	2576	2741	2965
Coverage (right class)	49	**55**	53	51	50	56	49	50	52	51
Coverage (neigh. class)	3	**4**	3	3	3	5	4	3	4	5
Precision (right class)	82	**68**	71	77	74	61	58	73	68	55
Precision (neigh. class)	18	**30**	28	23	25	39	39	25	31	41
JS-Distance (full set, class 1)	0.187	**0.182**	0.194	0.191	0.211	0.210	0.215	0.219	0.210	0.222
Unique rediscoveries	81	**71**	67	61	67	61	65	53	73	61
Learning rate	1e-6	**1e-5**	1e-2	1e-5	1e-5	1e-3	1e-5	1e-2	1e-5	1e-5
Batch size	100	**250**	250	1000	1000	100	250	100	500	100
Nodes per layer	1024	**512**	512	512	1024	256	1024	512	512	1024
Number of layers	8	**6**	8	7	5	6	8	6	6	8
Update steps	1.6e6	**8.0e5**	1.6e6	2.0e+05	8.0e5	8.0e5	1.6e6	8.0e5	8.0e5	1.6e6
Experiment ID	11	12	13	14	15	16	17	18	19	20
JS-Distance (training set, class 1)	0.285	0.286	0.288	0.288	0.292	0.292	0.292	0.295	0.297	0.297
Unique compositions	2819	2786	2987	2810	3670	3089	3079	2557	2322	3381
Coverage (right class)	52	48	54	52	56	52	53	48	46	53
Coverage (neigh. class)	4	4	4	3	4	5	4	3	3	3
Precision (right class)	73	70	74	76	63	64	64	81	67	69
Precision (neigh. class)	26	29	24	21	21	31	30	17	31	25
JS-Distance (full set, class 1)	0.206	0.244	0.241	0.228	0.243	0.236	0.212	0.260	0.269	0.226
Unique rediscoveries	55	53	69	55	78	54	52	53	53	56
Learning rate	1e-5	1e-3	1e-3	1e-3	1e-4	1e-3	1e-4	1e-4	1e-3	1e-4
Batch size	10	1000	10	100	100	500	1000	10	1000	2000
Nodes per layer	1024	128	32	256	512	256	512	32	128	1024
Number of layers	8	7	6	6	8	8	7	4	7	8
Update steps	8.0e5	4.0e5	8.0e5	2.0e5	8.0e4	4.0e5	8.0e4	1.2e6	1.2e6	8.0e4
Experiment ID	21	22	23	24	25	26	27	28	29	30
JS-Distance (training set, class 1)	0.304	0.307	0.308	0.311	0.314	0.315	0.317	0.320	0.323	0.323
Unique compositions	2490	1444	2801	1934	2108	2135	1959	1824	2230	2615
Coverage (right class)	38	32	52	37	42	39	44	41	42	47
Coverage (neigh. class)	3	1	3	2	3	2	2	2	3	2
Precision (right class)	69	85	81	73	73	74	93	82	68	70
56										
Precision (neigh. class)	29	15	16	27	24	18	7	18	31	19
JS-Distance (full set, class 1)	0.279	0.278	0.246	0.277	0.295	0.269	0.254	0.254	0.250	0.273
Unique rediscoveries	37	59	62	35	58	39	41	44	52	56
Learning rate	1e-3	1e-6	1e-2	1e-3	1e-4	1e-3	1e-6	1e-5	1e-2	1e-5
Batch size	2000	500	100	100	1000	500	10	500	100	10
Nodes per layer	32	512	512	32	32	128	1024	256	256	1024
Number of layers	8	8	6	4	2	4	7	6	5	8
Update steps	1.6e6	1.2e6	2.0e5	1.2e6	8.0e5	2.0e5	8.0e5	2.0e5	1.6e6	4.0e5
Experiment ID	31	32	33	34	35	36	37	38	39	40
JS-Distance (training set, class 1)	0.325	0.329	0.331	0.331	0.331	0.332	0.334	0.336	0.341	0.343
Unique compositions	1871	1931	2193	2503	2604	3270	1649	2576	2246	1741
Coverage (right class)	34	39	46	45	46	46	38	38	43	34
Coverage (neigh. class)	2	2	2	3	3	4	1	3	2	2
Precision (right class)	89	78	86	83	71	93	74	89	76	
Precision (neigh. class)	44	10	21	11	12	21	6	19	8	23
JS-Distance (full set, class 1)	0.299	0.290	0.245	0.283	0.285	0.287	0.263	0.324	0.300	0.307
Unique rediscoveries	42	40	42	55	56	71	32	59	50	48
Learning rate	1e-2	1e-3	1e-4	1e-6	1e-5	1e-4	1e-4	1e-4	1e-6	1e-2
Batch size	1000	1000	250	1000	10	10	250	2000	500	250
Nodes per layer	128	256	128	128	128	128	1024	32	256	256
Number of layers	8	3	5	4	3	6	3	2	5	6
Update steps	8.0e5	8.0e4	2.0e5	4.0e5	8.0e5	2.0e5	2.0e5	8.0e5	2.0e5	1.6e6

Table 7.4 Hyperparameter search for the RL. Chosen model highlighted in bold.

Table 7.5 Performance metrics for the three generative models conditioned on the minority class 1after 10,000 valid samples. Shown are averages over 50 model fits together with the standard deviation.

	RL	VAE	GAN
Precision (right class)	$69 \pm 8\%$	$83 \pm 1\%$	66 ± 2 %
Precision (neigh. class)	$30 \pm 7\%$	$16 \pm 1\%$	28 ± 1 %
Coverage (right class)	$53 \pm 3\%$	$54 \pm 2\%$	62 ± 1 %
Coverage (neigh. class)	$4 \pm 1\%$	$5 \pm 0\%$	10 ± 1 %
JS-Distance	0.20 ± 0.03	0.16 ± 0.01	0.16 ± 0.01

Table 7.6 Comparison of performance metrics for minority class 1 sample generation by the generative models and the substructure enrichment based sample selection. For equally ranked triples, random selection was performed 50 times and mean and standard deviation are reported.

	Enrich 25	Enrich 50	Enrich 100
Precision (right class)	$41 \pm 4\%$	$33 \pm 3\%$	$30 \pm 2\%$
Precision (neigh. class)	$37 \pm 3\%$	$35 \pm 3\%$	$35 \pm 2\%$
Coverage (right class)	$9 \pm 1\%$	$15 \pm 1\%$	$25 \pm 2\%$
Coverage (neigh. class)	$2 \pm 0\%$	$3 \pm 0\%$	$6 \pm 0\%$
JS-Distance	0.36 ± 0.03	0.34 ± 0.02	0.32 ± 0.01

Table 7.7 Performance metrics for the three generative models conditioned to generate majority class 5 compositions after 25 million valid samples. Shown are averages over 50 model fits together with the standard deviation.

	RL	VAE	GAN
Precision (right class)	$87 \pm 7\%$	$58 \pm 1\%$	54 ± 1 %
Precision (neigh. classes)	$13 \pm 7\%$	$34 \pm 1\%$	36 ± 1 %
Coverage (right class)	$44 \pm 15\%$	$86 \pm 9\%$	89 ± 2 %
Coverage (neigh. class)	$9 \pm 4\%$	$82 \pm 11\%$	85 ± 3 %
JS-Distance	0.49 ± 0.14	0.21 ± 0.01	0.16 ± 0.01

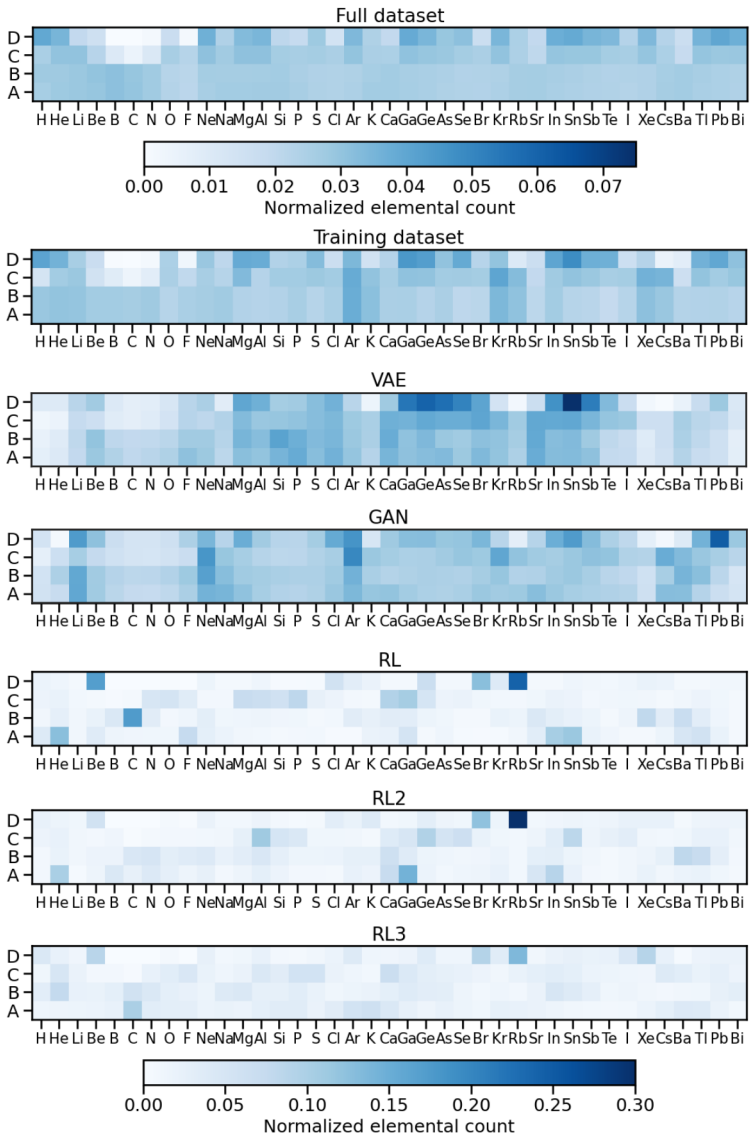


Figure 7.8 Top: Elemental distribution for class 5. For the RL model, results of 3 different model fits are shown to illustrate that different (local) minima are reached. A large divergence between RL and trainings dataset is clearly visible in these solutions, also explaining the overall lower space coverage produced by the RL model. The occurrence of such local minima is known [205, 206]. The could potentially be broadened by decreasing the hyperparameter β in the RL policy. Figure reproduced from [197] under CC BY 4.0.

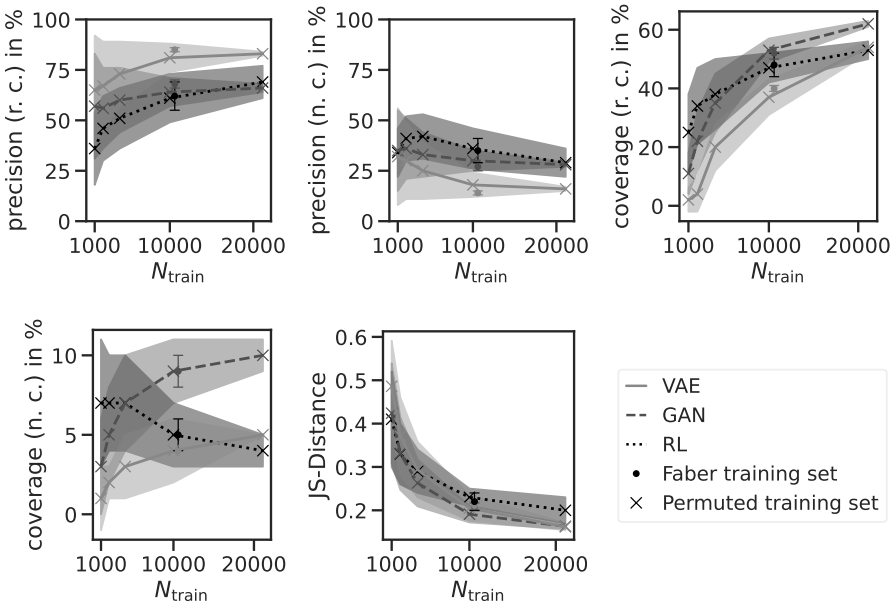


Figure 7.9 Influence of training set size on the performance metrics of VAE, GAN and RL models for minority class generation, after 10,000 generated samples. Shown are averages over 50 model fits employing 50 different random training subsets. Shaded areas represent the corresponding standard deviations. Figure reproduced from [197] under CC BY 4.0.

Summary and Conclusions

Many materials in the fields of energy conversion, electronics, catalysis and and magneto-optics are based on the double perovskite structure ABC_2D_6. Its four lattice sites can host numerous combinations of chemical elements leading to a large variety of properties. The simple crystal structure makes it an ideal candidate to highlight the relation between chemical composition and properties. Here we employed electronic structure calculation and different machine learning methods to explore the chemical space of double perovskites, with a special focus on photovoltaic applications.

In the first part we studied the power conversion efficiency, carriers effective mass and thermodynamic stability of halide double perovskites, with the aid of a regression model based on a convolutional neural network and the periodic table representation. Our work showed that there is a necessary trade-off between the properties that are desirable for an efficient and stable photovoltaic device.

In the second part we adopted a message passing neural network with explainable parameters to predict electronic energy levels in the bands, and the band gap of double perovskites. A principal component analysis followed by clustering of the network parameters showed that it is possible to group these materials in families with similar behavior based on what elements are present at the four lattice sites. However, the properties that the model uses to form the clusters are heavily influenced by the presence of a prior and its relative weight, and it is therefore fundamental to couple this method with a rigorous physical analysis.

The third part was dedicated to the inverse design of materials with the formation energy as a target property. The three generative models that were chosen for this task (VAE, GAN and RL) were able to propose many new compositions in the target property class with high precision and robustness, even with a reduced training set size. In particular, they showed a very promising performance when generating rare samples, which is highly desirable in data enhancement and materials discovery applications.

Bibliography

[1] J. Y. Kim, J.-W. Lee, H. S. Jung, H. Shin, N.-G. Park, *Chemical Reviews* **2020**, *120*, 7867–7918.

[2] V. Goldschmidt, *Berichte der Deutschen Chemischen Gesellschaft* **1927**, *60*, 1263–1296.

[3] C. Li, X. Lu, W. Ding, L. Feng, Y. Gao, Z. Guo, *Acta Crystallographica Section B: Structural Science* **2008**, *64*, 702–707.

[4] G. Pacchioni, *Nature Reviews Materials* **2021**, *6*, 108–108.

[5] X. Zhang, W. Wang, B. Xu, S. Liu, H. Dai, D. Bian, S. Chen, K. Wang, X. W. Sun, *Nano Energy* **2017**, *37*, 40–45.

[6] K. Lin, J. Xing, L. N. Quan, F. P. G. de Arquer, X. Gong, J. Lu, L. Xie, W. Zhao, D. Zhang, C. Yan, et al., *Nature* **2018**, *562*, 245–248.

[7] W. Wang, M. Xu, X. Xu, W. Zhou, Z. Shao, *Angewandte Chemie International Edition* **2020**, *59*, 136–152.

[8] M. A. B. Adnan, K. Arifin, L. J. Minggu, M. B. Kassim, *International Journal of Hydrogen Energy* **2018**, *43*, 23209–23220.

[9] W. Tian, H. Zhou, L. Li, *Small* **2017**, *13*, 1702107.

[10] L. Dou, Y. Yang, J. You, Z. Hong, W.-H. Chang, G. Li, Y. Yang, *Nature Communications* **2014**, *5*, 5404.

[11] L. Shu, J. Sunarso, S. S. Hashim, J. Mao, W. Zhou, F. Liang, *International Journal of Hydrogen Energy* **2019**, *44*, 31275–31304.

[12] S. Tao, J. T. Irvine, J. A. Kilner, *Advanced Materials* **2005**, *17*, 1734–1737.

[13] B. R. Sutherland, E. H. Sargent, *Nature Photonics* **2016**, *10*, 295–302.

[14] H. Zhu, Y. Fu, F. Meng, X. Wu, Z. Gong, Q. Ding, M. V. Gustafsson, M. T. Trinh, S. Jin, X. Zhu, *Nature Materials* **2015**, *14*, 636–642.

[15] J. Höcker, F. Brust, M. Armer, V. Dyakonov, *CrystEngComm* **2021**, *23*, 2202–2207.

[16] A. Biewald, N. Giesbrecht, T. Bein, P. Docampo, A. Hartschuh, R. Ciesielski, *ACS Applied Materials & Interfaces* **2019**, *11*, 20838–20844.

[17] A. M. Leguy, Y. Hu, M. Campoy-Quiles, M. I. Alonso, O. J. Weber, P. Azarhoosh, M. Van Schilfgaarde, M. T. Weller, T. Bein, J. Nelson, et al., *Chemistry of Materials* **2015**, *27*, 3397–3407.

[18] W. Tress, *Advanced Energy Materials* **2017**, *7*, 1602358.

[19] Y. Zhao, K. Zhu, *Chemical Society Reviews* **2016**, *45*, 655–689.

[20] T. Zhang, M. Wang, H. Yang, *Energies* **2018**, *11*, 3157.

[21] A. Kojima, K. Teshima, Y. Shirai, T. Miyasaka, *Journal of the American Chemical Society* **2009**, *131*, 6050–6051.

[22] H.-S. Kim, C.-R. Lee, J.-H. Im, K.-B. Lee, T. Moehl, A. Marchioro, S.-J. Moon, R. Humphry-Baker, J.-H. Yum, J. E. Moser, et al., *Scientific Reports* **2012**, *2*, 1–7.

[23] N. Giesbrecht, J. Schlipf, L. Oesinghaus, A. Binek, T. Bein, P. Müller-Buschbaum, P. Docampo, *ACS Energy Letters* **2016**, *1*, 150–154.

[24] M. Green, E. Dunlop, J. Hohl-Ebinger, M. Yoshita, N. Kopidakis, X. Hao, *Progress in Photovoltaics: Research and Applications* **2021**, *29*, 3–15.

[25] Best Research-Cell Efficiency Chart, https://www.nrel.gov/pv/cell-efficiency.html.

[26] S. A. Mhamad, A. M. Mohammed, M. Aziz, F. Aziz in *Nanostructured Materials for Next-Generation Energy Storage and Conversion: Photovoltaic and Solar Energy*, Springer, **2019**, pp. 227–246.

[27] P. Roy, A. Ghosh, F. Barclay, A. Khare, E. Cuce, *Coatings* **2022**, *12*, 1089.

[28] Y. Bi, E. M. Hutter, Y. Fang, Q. Dong, J. Huang, T. J. Savenije, *The Journal of Physical Chemistry Letters* **2016**, *7*, 923–928.

[29] T. Dittrich, F. Lang, O. Shargaieva, J. Rappich, N. Nickel, E. Unger, B. Rech, *Applied Physics Letters* **2016**, *109*, 073901.

[30] A. Goyal, S. McKechnie, D. Pashov, W. Tumas, M. Van Schilfgaarde, V. Stevanovic, *Chemistry of Materials* **2018**, *30*, 3920–3928.

[31] J. Feng, B. Xiao, *The Journal of Physical Chemistry Letters* **2014**, *5*, 1278–1282.

[32] R. A. Jishi, O. B. Ta, A. A. Sharif, *The Journal of Physical Chemistry C* **2014**, *118*, 28344–28349.

[33] K. T. Butler, J. M. Frost, A. Walsh, *Materials Horizons* **2015**, *2*, 228–231.

[34] T. Wang, B. Daiber, J. M. Frost, S. A. Mann, E. C. Garnett, A. Walsh, B. Ehrler, *Energy & Environmental Science* **2017**, *10*, 509–515.

[35] M. Baranowski, P. Plochocka, *Advanced Energy Materials* **2020**, *10*, 1903659.

[36] J. M. Frost, A. Walsh, *Accounts of Chemical Research* **2016**, *49*, 528–535.

[37] E. Menéndez-Proupin, C. L. Beltrán Rıos, P. Wahnón, *Physica Status Solidi (RRL)–Rapid Research Letters* **2015**, *9*, 559–563.

[38] C. Wehrenfennig, G. E. Eperon, M. B. Johnston, H. J. Snaith, L. M. Herz, *Advanced Materials* **2014**, *26*, 1584–1589.

[39] S. G. Motti, D. Meggiolaro, S. Martani, R. Sorrentino, A. J. Barker, F. De Angelis, A. Petrozza, *Advanced Materials* **2019**, *31*, 1901183.

[40] T. S. Sherkar, C. Momblona, L. Gil-Escrig, J. Avila, M. Sessolo, H. J. Bolink, L. J. A. Koster, *ACS Energy Letters* **2017**, *2*, 1214–1222.

[41] C. C. Stoumpos, C. D. Malliakas, M. G. Kanatzidis, *Inorganic Chemistry* **2013**, *52*, 9019–9038.

[42] J. Cao, F. Yan, *Energy & Environmental Science* **2021**, *14*, 1286–1325.

[43] X. Jiang, Z. Zang, Y. Zhou, H. Li, Q. Wei, Z. Ning, *Accounts of Materials Research* **2021**, *2*, 210–219.

[44] T. Krishnamoorthy, H. Ding, C. Yan, W. L. Leong, T. Baikie, Z. Zhang, M. Sherburne, S. Li, M. Asta, N. Mathews, et al., *Journal of Materials Chemistry A* **2015**, *3*, 23829–23832.

[45] Y.-T. Huang, S. R. Kavanagh, D. O. Scanlon, A. Walsh, R. L. Hoye, *Nanotechnology* **2021**, *32*, 132004.

[46] I. Kopacic, B. Friesenbichler, S. F. Hoefler, B. Kunert, H. Plank, T. Rath, G. Trimmel, *ACS Applied Energy Materials* **2018**, *1*, 343–347.

[47] J. Chang, L. Jiang, G. Wang, W. Zhao, Y. Huang, H. Chen, *Physical Chemistry Chemical Physics* **2021**, *23*, 14449–14456.

[48] C. J. Bartel, C. Sutton, B. R. Goldsmith, R. Ouyang, C. B. Musgrave, L. M. Ghiringhelli, M. Scheffler, *Science Advances* **2019**, *5*, eaav0693.

[49] X. Yang, W. Wang, R. Ran, W. Zhou, Z. Shao, *Energy & Fuels* **2020**, *34*, 10513–10528.

[50] A. H. Slavney, L. Leppert, D. Bartesaghi, A. Gold-Parker, M. F. Toney, T. J. Savenije, J. B. Neaton, H. I. Karunadasa, *Journal of the American Chemical Society* **2017**, *139*, 5015–5018.

[51] E. T. McClure, M. R. Ball, W. Windl, P. M. Woodward, *Chemistry of Materials* **2016**, *28*, 1348–1354.

[52] M. T. Sirtl, M. Armer, L. K. Reb, R. Hooijer, P. Dörflinger, M. A. Scheel, K. Tvingstedt, P. Rieder, N. Glück, P. Pandit, S. V. Roth, P. Müller-Buschbaum, V. Dyakonov, T. Bein, *ACS Applied Energy Materials* **2020**, *3*, 11597–11609.

[53] X. Yang, Y. Chen, P. Liu, H. Xiang, W. Wang, R. Ran, W. Zhou, Z. Shao, *Advanced Functional Materials* **2020**, *30*, 2001557.

[54] M. Armer, J. Höcker, C. Büchner, S. Häfele, P. Dörflinger, M. T. Sirtl, K. Tvingstedt, T. Bein, V. Dyakonov, *CrystEngComm* **2021**, *23*, 6848–6854.

[55] M. R. Filip, S. Hillman, A. A. Haghighirad, H. J. Snaith, F. Giustino, *The Journal of Physical Chemistry Letters* **2016**, *7*, 2579–2585.

[56] R. L. Hoye, L. Eyre, F. Wei, F. Brivio, A. Sadhanala, S. Sun, W. Li, K. H. Zhang, J. L. MacManus-Driscoll, P. D. Bristowe, et al., *Advanced Materials Interfaces* **2018**, *5*, 1800464.

[57] W. Tress, M. T. Sirtl, *Solar RRL* **2022**, *6*, 2100770.

[58] J. Luo, S. Li, H. Wu, Y. Zhou, Y. Li, J. Liu, J. Li, K. Li, F. Yi, G. Niu, et al., *ACS Photonics* **2018**, *5*, 398–405.

[59] J. C. Dahl, W. T. Osowiecki, Y. Cai, J. K. Swabeck, Y. Bekenstein, M. Asta, E. M. Chan, A. P. Alivisatos, *Chemistry of Materials* **2019**, *31*, 3134–3143.

[60] W. Meng, X. Wang, Z. Xiao, J. Wang, D. B. Mitzi, Y. Yan, *The Journal of Physical Chemistry Letters* **2017**, *8*, 2999–3007.

[61] A. H. Slavney, L. Leppert, A. Saldivar Valdes, D. Bartesaghi, T. J. Savenije, J. B. Neaton, H. I. Karunadasa, *Angewandte Chemie International Edition* **2018**, *57*, 12765–12770.

[62] G. M. Dalpian, X.-G. Zhao, L. Kazmerski, A. Zunger, *Chemistry of Materials* **2019**, *31*, 2497–2506.

[63] Z. Li, S. R. Kavanagh, M. Napari, R. G. Palgrave, M. Abdi-Jalebi, Z. Andaji-Garmaroudi, D. W. Davies, M. Laitinen, J. Julin, M. A. Isaacs, et al., *Journal of Materials Chemistry A* **2020**, *8*, 21780–21788.

[64] K.-z. Du, W. Meng, X. Wang, Y. Yan, D. B. Mitzi, *Angewandte Chemie International Edition* **2017**, *56*, 8158–8162.

[65] H. Wu, A. Erbing, M. B. Johansson, J. Wang, C. Kamal, M. Odelius, E. M. Johansson, *ChemSusChem* **2021**, *14*, 4507–4515.

[66] J. F. Silveira, J. L. Da Silva, *ACS Applied Energy Materials* **2020**, *3*, 7364–7371.

[67] N. R. Wolf, B. A. Connor, A. H. Slavney, H. I. Karunadasa, *Angewandte Chemie* **2021**, *133*, 16400–16414.

[68] T. T. Tran, J. R. Panella, J. R. Chamorro, J. R. Morey, T. M. McQueen, *Materials Horizons* **2017**, *4*, 688–693.

[69] Q. Li, Y. Wang, W. Pan, W. Yang, B. Zou, J. Tang, Z. Quan, *Angewandte Chemie International Edition* **2017**, *56*, 15969–15973.

[70] J. Yang, P. Zhang, S.-H. Wei, *The Journal of Physical Chemistry Letters* **2018**, *9*, 31–35.

[71] F. Ji, J. Klarbring, F. Wang, W. Ning, L. Wang, C. Yin, J. S. M. Figueroa, C. K. Christensen, M. Etter, T. Ederth, et al., *Angewandte Chemie* **2020**, *132*, 15303–15306.

[72] R. Jacobs, G. Luo, D. Morgan, *Advanced Functional Materials* **2019**, *29*, 1804354.

[73] S. X. Tao, X. Cao, P. A. Bobbert, *Scientific Reports* **2017**, *7*, 1–9.

[74] G. Volonakis, M. R. Filip, A. A. Haghighirad, N. Sakai, B. Wenger, H. J. Snaith, F. Giustino, *The Journal of Physical Chemistry Letters* **2016**, *7*, 1254–1259.

[75] M. R. Filip, X. Liu, A. Miglio, G. Hautier, F. Giustino, *The Journal of Physical Chemistry C* **2018**, *122*, 158–170.

[76] J. Dai, L. Ma, M. Ju, J. Huang, X. C. Zeng, *Physical Chemistry Chemical Physics: PCCP* **2017**, *19*, 21691–21695.

[77] X.-G. Zhao, J.-H. Yang, Y. Fu, D. Yang, Q. Xu, L. Yu, S.-H. Wei, L. Zhang, *Journal of the American Chemical Society* **2017**, *139*, 2630–2638.

[78] Z. Xiao, K.-Z. Du, W. Meng, J. Wang, D. B. Mitzi, Y. Yan, *Journal of the American Chemical Society* **2017**, *139*, 6054–6057.

[79] G. Volonakis, A. A. Haghighirad, H. J. Snaith, F. Giustino, *The Journal of Physical Chemistry Letters* **2017**, *8*, 3917–3924.

[80] M. Roknuzzaman, C. Zhang, K. K. Ostrikov, A. Du, H. Wang, L. Wang, T. Tesfamichael, *Scientific Reports* **2019**, *9*, 1–7.

[81] X.-L. Ding, Z. Gao, G. Mao, S. Chen, Y. Bai, P. Gao, I. D. Gates, C. Wu, W. Yang, J. Yao, *Physical Chemistry Chemical Physics* **2022**, *24*, 3460.

[82] Y. Cai, W. Xie, Y. T. Teng, P. C. Harikesh, B. Ghosh, P. Huck, K. A. Persson, N. Mathews, S. G. Mhaisalkar, M. Sherburne, et al., *Chemistry of Materials* **2019**, *31*, 5392–5401.

[83] C. J. Bartel, J. M. Clary, C. Sutton, D. Vigil-Fowler, B. R. Goldsmith, A. M. Holder, C. B. Musgrave, *Journal of the American Chemical Society* **2020**, *142*, 5135–5145.

[84] A. Szabo, N. S. Ostlund, *Modern quantum chemistry: introduction to advanced electronic structure theory*, Courier Corporation, **2012**.

[85] C. J. Cramer, *Essentials of computational chemistry: theories and models*, John Wiley & Sons, **2013**.

[86] T. Etienne, Molecular electronic-structure theory, https://thibaudetienne.wordpress.com/, **2022**.

[87] P. Hohenberg, W. Kohn, *Physical Review* **1964**, *136*, B864.

[88] W. Kohn, L. J. Sham, *Physical Review* **1965**, *140*, A1133.

[89] J. Als-Nielsen, D. McMorrow, *Elements of modern X-ray physics*, John Wiley & Sons, **2011**.

[90] C. Kittel, **2021**.

[91] N. W. Ashcroft, N. D. Mermin, *Solid state physics*, Cengage Learning, **2022**.

[92] J. J. Quinn, K.-S. Yi, *Solid state physics: principles and modern applications*, Springer Science & Business Media, **2009**.

[93] I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning*, http://www.deeplearningbook.org, MIT Press, **2016**.

[94] A. Zhang, Z. C. Lipton, M. Li, A. J. Smola, *arXiv preprint arXiv:2106.11342* **2021**.

[95] Y. LeCun, Y. Bengio, et al., *The Handbook of Brain Theory and Neural Networks* **1995**, *3361*, 1995.

[96] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, G. E. Dahl in International Conference on Machine Learning, PMLR, **2017**, pp. 1263–1272.

[97] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, *Advances in Neural Information Processing Systems* **2017**, *30*.

[98] M.-T. Luong, H. Pham, C. D. Manning, *arXiv preprint arXiv:1508.04025* **2015**.

[99] D. Bahdanau, K. Cho, Y. Bengio, *arXiv preprint arXiv:1409.0473* **2014**.

[100] A. Brock, J. Donahue, K. Simonyan, *arXiv preprint arXiv:1809.11096* **2019**.

[101] W. Nie, N. Narodytska, A. Patel in International Conference on Learning Representations, **2019**.

[102] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, Y.-H. Yang in Proceedings of the AAAI Conf. on Artif. Intell. *Vol. 32*, **2018**.

[103] D. C. Elton, Z. Boukouvalas, M. D. Fuge, P. W. Chung, *Molecular Systems Design & Engineering* **2019**, *4*, 828–849.

[104] K. M. Merz Jr, G. De Fabritiis, G. Wei, JCIM special issue on generative models for molecular design, **2020**.

[105] J. Noh, G. H. Gu, S. Kim, Y. Jung, *Chemical Science* **2020**, *11*, 4871–4881.

[106] B. Sanchez-Lengeling, A. Aspuru-Guzik, *Science* **2018**, *361*, 360–365.

[107] D. Merk, L. Friedrich, F. Grisoni, G. Schneider, *Molecular Informatics* **2018**, *37*, 1700153.

[108] R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, A. Aspuru-Guzik, *ACS Central Science* **2018**, *4*, 268–276.

[109] W. Jin, R. Barzilay, T. Jaakkola in International Conference on Machine Learning, PMLR, **2018**, pp. 2323–2332.

[110] R.-R. Griffiths, J. M. Hernández-Lobato, *Chemical Science* **2020**, *11*, 577–586.

[111] D. Weininger, *Journal of Chemical Information and Computer Sciences* **1988**, *28*, 31–36.

[112] M. Krenn, F. Häse, A. Nigam, P. Friederich, A. Aspuru-Guzik, *Machine Learning: Science and Technology* **2020**, *1*, 045024.

[113] Y. Li, L. Zhang, Z. Liu, *Journal of Cheminformatics* **2018**, *10*, 1–24.

[114] L. Chen, W. Zhang, Z. Nie, S. Li, F. Pan, *Journal of Materials Informatics* **2021**, *1*, 4.

[115] F. A. Faber, A. Lindmaa, O. A. Von Lilienfeld, R. Armiento, *Physical Review Letters* **2016**, *117*, 135502.

[116] Y. Pathak, K. S. Juneja, G. Varma, M. Ehara, U. D. Priyakumar, *Physical Chemistry Chemical Physics* **2020**, *22*, 26935–26943.

[117] A. Nouira, N. Sokolovska, J.-C. Crivello, *arXiv preprint arXiv:1810.11203* **2018**.

[118] Z. Ren, S. I. P. Tian, J. Noh, F. Oviedo, G. Xing, J. Li, Q. Liang, R. Zhu, A. G. Aberle, S. Sun, et al., *Matter* **2022**, *5*, 314–335.

[119] J. Noh, J. Kim, H. S. Stein, B. Sanchez-Lengeling, J. M. Gregoire, A. Aspuru-Guzik, Y. Jung, *Matter* **2019**, *1*, 1370–1384.

[120] Z. Yao, B. Sánchez-Lengeling, N. S. Bobbitt, B. J. Bucior, S. G. H. Kumar, S. P. Collins, T. Burns, T. K. Woo, O. K. Farha, R. Q. Snurr, et al., *Nature Machine Intelligence* **2021**, *3*, 76–86.

[121] T. Xie, J. C. Grossman, *Physical Review Letters* **2018**, *120*, 145301.

[122] T. Long, N. M. Fortunato, I. Opahle, Y. Zhang, I. Samathrakis, C. Shen, O. Gutfleisch, H. Zhang, *npj Computational Materials* **2021**, *7*, 1–7.

[123] S. Kim, J. Noh, G. H. Gu, A. Aspuru-Guzik, Y. Jung, *ACS Central Science* **2020**, *6*, 1412–1420.

[124] Y. Dan, Y. Zhao, X. Li, S. Li, M. Hu, J. Hu, *npj Computational Materials* **2020**, *6*, 1–7.

[125] Y. Zhao, M. Al-Fahdi, M. Hu, E. M. Siriwardane, Y. Song, A. Nasiri, J. Hu, *Advanced Science* **2021**, *8*, 2100566.

[126] A. Debnath, A. M. Krajewski, H. Sun, S. Lin, M. Ahn, W. Li, S. Priya, J. Singh, S. Shang, A. M. Beese, et al., *arXiv preprint arXiv:2108.12019* **2021**.

[127] C. J. Court, B. Yildirim, A. Jain, J. M. Cole, *Journal of Chemical Information and Modeling* **2020**, *60*, 4518–4535.

[128] I.-H. Lee, K. J. Chang, *Computational Materials Science* **2021**, *194*, 110436.

[129] J. Hoffmann, L. Maestrati, Y. Sawada, J. Tang, J. M. Sellier, Y. Bengio, *arXiv preprint arXiv:1909.00949* **2019**.

[130] V. Korolev, A. Mitrofanov, A. Eliseev, V. Tkachenko, *Materials Horizons* **2020**, *7*, 2710–2718.

[131] I. Sajedian, T. Badloe, J. Rho, *Optics Express* **2019**, *27*, 5874–5883.

[132] C. Luo, S. Ning, Z. Liu, Z. Zhuang, *Extreme Mechanics Letters* **2020**, *36*, 100651.

[133] D. P. Kingma, M. Welling, *arXiv preprint arXiv:1312.6114* **2013**.

[134] D. P. Kingma, M. Welling, et al., *Foundations and Trends® in Machine Learning* **2019**, *12*, 307–392.

[135] D. J. Rezende, S. Mohamed, D. Wierstra in Proceedings of the 31st International Conference on Machine Learning, PMLR 32(2), **2014**, pp. 1278–1286.

[136] J. Lucas, G. Tucker, R. B. Grosse, M. Norouzi in DGS@ICLR, **2019**.

[137] M. J. Kusner, B. Paige, J. M. Hernández-Lobato in International conference on machine learning, PMLR, **2017**, pp. 1945–1954.

[138] O. Ivanov, M. Figurnov, D. Vetrov, *arXiv preprint arXiv:1806.02382* **2018**.

[139] J. Lim, S. Ryu, J. W. Kim, W. Y. Kim, *Journal of Cheminformatics* **2018**, *10*, 1–9.

[140] M. Lotfollahi, M. Naghipourfar, F. J. Theis, F. A. Wolf, *arXiv preprint arXiv:1910.01791* **2019**.

[141] A. Mollaysa, B. Paige, A. Kalousis, **2019**.

[142] K. Sohn, H. Lee, X. Yan, *Advances in Neural Information Processing Systems* **2015**, *28*.

[143] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, *arXiv preprint arXiv:1406.2661* **2014**.

[144] L. Metz, B. Poole, D. Pfau, J. Sohl-Dickstein, *arXiv preprint arXiv:1611.02163* **2017**.

[145] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, *arXiv preprint arXiv:1606.03498* **2016**.

[146] J. Gui, Z. Sun, Y. Wen, D. Tao, J. Ye, *arXiv preprint arXiv:2001.06937* **2020**.

[147] M. Arjovsky, S. Chintala, L. Bottou in International Conference on Machine Learning, PMLR, **2017**, pp. 214–223.

[148] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. C. Courville, *Advances in Neural Information Processing Systems* **2017**, *30*.

[149] J. Kober, J. A. Bagnell, J. Peters, *The International Journal of Robotics Research* **2013**, *32*, 1238–1274.

[150] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, *arXiv preprint arXiv:1312.5602* **2013**.

[151] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al., *Nature* **2016**, *529*, 484–489.

[152] J. Schmidt, J. Shi, P. Borlido, L. Chen, S. Botti, M. A. Marques, *Chemistry of Materials* **2017**, *29*, 5090–5103.

[153] W. A. Saidi, W. Shadid, I. E. Castelli, *npj Computational Materials* **2020**, *6*, 1–7.

[154] C. Li, H. Hao, B. Xu, G. Zhao, L. Chen, S. Zhang, H. Liu, *Journal of Materials Chemistry C* **2020**, *8*, 3127–3136.

[155] G. Pilania, A. Mannodi-Kanakkithodi, B. Uberuaga, R. Ramprasad, J. Gubernatis, T. Lookman, *Scientific Reports* **2016**, *6*, 1–10.

[156] M. Kuisma, J. Ojanen, J. Enkovaara, T. T. Rantala, *Physical Review B* **2010**, *82*, 115106.

[157] L. J. Nelson, G. L. Hart, F. Zhou, V. Ozoliņš, et al., *Physical Review B* **2013**, *87*, 035125.

[158] M. Agiorgousis, Y.-Y. Sun, D.-H. Choe, D. West, S. Zhang, *Advanced Theory and Simulations* **2019**, *2*, 1800173.

[159] J. Im, S. Lee, T.-W. Ko, H. W. Kim, Y. Hyon, H. Chang, *npj Computational Materials* **2019**, *5*, 1–8.

[160] Z. Yang, Y. Liu, Y. Zhang, L. Wang, C. Lin, Y. Lv, Y. Ma, C. Shao, *The Journal of Physical Chemistry C* **2021**, *125*, 22483.

[161] T. Konno, *Journal of the Physical Society of Japan* **2020**, *89*, 124006.

[162] X. Zheng, P. Zheng, R.-Z. Zhang, *Chemical Science* **2018**, *9*, 8426–8432.

[163] T. Konno, H. Kurokawa, F. Nabeshima, Y. Sakishita, R. Ogawa, I. Hosako, A. Maeda, *Physical Review B* **2021**, *103*, 014509.

[164] A. L. Maas, A. Y. Hannun, A. Y. Ng, et al. in Proc. icml, *Vol. 30*, Citeseer, **2013**, p. 3.

[165] J. L. Ba, J. R. Kiros, G. E. Hinton, *arXiv preprint arXiv:1607.06450* **2016**.

[166] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R. R. Salakhutdinov, *arXiv preprint arXiv:1207.0580* **2012**.

[167] D. P. Kingma, J. Ba, *arXiv preprint arXiv:1412.6980* **2014**.

[168] I. Loshchilov, F. Hutter, *arXiv preprint arXiv:1608.03983* **2016**.

[169] V. Blum, R. Gehrke, F. Hanke, P. Havu, V. Havu, X. Ren, K. Reuter, M. Scheffler, *Computer Physics Communications* **2009**, *180*, 2175–2196.

[170] J. Heyd, G. E. Scuseria, M. Ernzerhof, *The Journal of Chemical Physics* **2003**, *118*, 8207–8215.

[171] A. V. Krukau, O. A. Vydrov, A. F. Izmaylov, G. E. Scuseria, *The Journal of Chemical Physics* **2006**, *125*, 224106.

[172] W. P. Huhn, V. Blum, *Physical Review Materials* **2017**, *1*, 033803.

[173] W. Pu, W. Xiao, J.-W. Wang, X.-W. Li, L. Wang, *Mater. Des.* **2021**, *198*, 109387.

[174] L. Yu, A. Zunger, *Physical Review Letters* **2012**, *108*, 068701.

[175] Reference Solar Spectral Irradiance: Air Mass 1.5, https://www.nrel.gov/grid/solar-resource/spectra-am1.5.html.

[176] B. Blank, T. Kirchartz, S. Lany, U. Rau, *Physical Review Applied* **2017**, *8*, 024032.

[177] W. Sun, S. T. Dacek, S. P. Ong, G. Hautier, A. Jain, W. D. Richards, A. C. Gamst, K. A. Persson, G. Ceder, *Science Advances* **2016**, *2*, e1600225.

[178] B. D. Pelatt, R. Ravichandran, J. F. Wager, D. A. Keszler, *Journal of the American Chemical Society* **2011**, *133*, 16852–16860.

[179] C. G. Van de Walle, J. Neugebauer, *Nature* **2003**, *423*, 626–628.

[180] B. D. Pelatt, R. S. Kokenyesi, R. Ravichandran, C. B. Pereira, J. F. Wager, D. A. Keszler, *Journal of Solid State Chemistry* **2015**, *231*, 138–144.

[181] D. W. Davies, K. T. Butler, A. J. Jackson, A. Morris, J. M. Frost, J. M. Skelton, A. Walsh, *Chem* **2016**, *1*, 617–627.

[182] R. E. Goodall, A. A. Lee, *Nature Communications* **2020**, *11*, 6280.

[183] A. Y.-T. Wang, S. K. Kauwe, R. J. Murdock, T. D. Sparks, *npj Computational Materials* **2021**, *7*, 77.

[184] A. Ihalage, Y. Hao, *Advanced Science* **2022**, *9*, 2200164.

[185] P. Schwaller, B. Hoover, J.-L. Reymond, H. Strobelt, T. Laino, *Science Advances* **2021**, *7*, eabe4166.

[186] A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, et al., *APL Materials* **2013**, *1*, 011002.

[187] J. M. Munro, K. Latimer, M. K. Horton, S. Dwaraknath, K. A. Persson, *npj Computational Materials* **2020**, *6*, 112.

[188] A. Jain, G. Hautier, S. P. Ong, C. J. Moore, C. C. Fischer, K. A. Persson, G. Ceder, *Physical Review B* **2011**, *84*, 045115.

[189] A. Wang, R. Kingsbury, M. McDermott, M. Horton, A. Jain, S. P. Ong, S. Dwaraknath, K. A. Persson, *Scientific Reports* **2021**, *11*, 1–10.

[190] M. Aykol, S. S. Dwaraknath, W. Sun, K. A. Persson, *Science Advances* **2018**, *4*, eaaq0148.

[191] J.-P. Desclaux, *Atomic data and nuclear data tables* **1973**, *12*, 311–406.

[192] D. Polykovskiy, A. Zhebrak, B. Sanchez-Lengeling, S. Golovanov, O. Tatanov, S. Belyaev, R. Kurbanov, A. Artamonov, V. Aladinskiy, M. Veselov, et al., *Frontiers in Pharmacology* **2020**, *11*, 565644.

[193] J. Zhang, R. Mercado, O. Engkvist, H. Chen, *Journal of Chemical Information and Modeling* **2021**, *61*, 2572–2581.

[194] N. Brown, M. Fiscato, M. H. Segler, A. C. Vaucher, *Journal of Chemical Information and Modeling* **2019**, *59*, 1096–1108.

[195] P. Renz, D. Van Rompaey, J. K. Wegner, S. Hochreiter, G. Klambauer, *Drug Discovery Today: Technologies* **2019**, *32*, 55–63.

[196] Y. Sawada, K. Morikawa, M. Fujii, *arXiv preprint arXiv:1910.11499* **2019**.

[197] H. Türk, E. Landini, C. Kunkel, J. T. Margraf, K. Reuter, *Chemistry of Materials* **2022**, *34*, 9455–9467.

[198] H. Huo, M. Rupp, *Machine Learning: Science and Technology* **2022**, *3*, 045017.

[199] M. J. Willatt, F. Musil, M. Ceriotti, *Physical Chemistry Chemical Physics* **2018**, *20*, 29661–29668.

[200] Q. Zhou, P. Tang, S. Liu, J. Pan, Q. Yan, S.-C. Zhang, *Proceedings of the National Academy of Sciences* **2018**, *115*, E6411–E6417.

[201] X. Zheng, P. Zheng, L. Zheng, Y. Zhang, R.-Z. Zhang, *Computational Materials Science* **2020**, *173*, 109436.

[202] V. Tshitoyan, J. Dagdelen, L. Weston, A. Dunn, Z. Rong, O. Kononova, K. A. Persson, G. Ceder, A. Jain, *Nature* **2019**, *571*, 95–98.

[203] C. Yan, S. Wang, J. Yang, T. Xu, J. Huang in Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics, **2020**, pp. 1–7.

[204] R. Amit, R. Meir, K. Ciosek in International conference on machine learning, PMLR, **2020**, pp. 269–278.

[205] S. S. Mousavi, M. Schukat, E. Howley in Proceedings of SAI Intelligent Systems Conference (IntelliSys) 2016: Volume 2, Springer, **2018**, pp. 426–440.

[206] L. A. Thiede, M. Krenn, A. Nigam, A. Aspuru-Guzik, *Machine Learning: Science and Technology* **2022**, *3*, 035008.

List of Abbreviations

AIMD *ab initio* Molecular Dynamics. 11

BSE Bethe-Salpeter Equation. 12

CBM Conduction Band Minimum. 4

CIGS Copper Indium Gallium Selenide. 2

CNN Convolutional Neural Network. 30

DFT Density Functional Theory. 11, 15

ETL Electron Transport Layer. 4

FAPI Formamidinium Lead Iodide. 2

FTO Fluorine-doped Tin Oxide. 4

GAN Generative Adversarial Network. 36

GBRT Gradient Boosted Regression Tree. 41

GGA Generalized Gradient Approximation. 16

GPR Gaussian Process Regression. 41

HK Hohenberg-Kohn. 15

HTL Hole Transport Layer. 4

ITO Indium Tin Oxide. 4

JS Jensen-Shannon. 78

KL Kullback-Leibler. 37

KRR Kernel Ridge Regression. 41

KS Kohn-Sham. 16

LDA Local Density Approximation. 16

LLSF Linear Least Square Fit. 42

MAE Mean Absolute Error. 75

MAPI Methylammonium Lead Iodide. 2

MSGD Minibatch Stochastic Gradient Descent. 28

PCA Principal Component Analysis. 59

PCE Power Conversion Efficiency. 2

PTR Periodic Table Representation. 42

RL Reinforcement Learning. 36

RMSE Root Mean Squared Error. 42

RNN Recurrent Neural Network. 36

SELFIES Self-Referencing Embedded Strings. 35

SISSO Sure Independence Screening and Sparsifying Operator. 8

SLME Spectroscopic Limited Maximum Efficiency. 44

SMILES Simplified Molecular Input Line Entry System. 35

SOC Spin-Orbit Coupling. 6

SRH Shockley-Read-Hall. 6

SSE Solid State Energy. 53

VAE Variational Autoencoder. 36

VBM Valence Band Maximum. 4

WCSSC Within-Cluster Sum-of-Squares Criterion. 59

WGAN Wasserstein GAN. 38