Master's Thesis Project

A Study of the Role of Interfaces in the Degradation of Lead-Tin Perovskite Solar Cells

Fiona Alice McPartlan

Student ID: 14154552

A thesis submitted in part fulfilment of the degree of

MSc. in Physics & Astronomy: Science for Energy and Sustainability

Supervisors: Prof. Dr. Bruno Ehrler & Prof. Dr. Esther Alarcón Lladó

UNIVERSITEIT VAN AMSTERDAM

VRIJE UNIVERSITEIT AMSTERDAM

School of Physics

University of Amsterdam and Vrije Universiteit Amsterdam

July 18, 2024

Table of Contents

Abstract

Despite being a promising material that can produce high-efficiency, solar cells with attractive, tunable bandgaps in the 1.2 - 1.5 eV range, lead-tin perovskite faces an ongoing issue with stability. Interfacial layers in perovskite solar cells can contribute to their degradation. In this thesis, we will examine the influence of the metal electrode/electron transport layer (C_{60} and BCP)/perovskite interfaces on the degradation of planar *p-i-n* structure $CsFAPbSnI_3$ perovskite solar cells.

The samples were fabricated using a SCIPRIOS SpinBot in order to improve the reproducibility of full and partial stack samples within and between batches. The manual spincoating recipe was successfully optimised for the SpinBot to ensure higher quality samples for this study. Examination on the degradation of the fresh and aged samples was done through analysis of UV-Vis absorbance, XRD patterns, J-V curve measurements, SEM imaging, and EDX scans.

It was found that the interface between the electron transport layer and the perovskite has a large influence on the operational stability of the perovskite solar cell devices and changes in the morphology of the perovskite surface. After ageing, white rods were observed on the surface of the perovskite. These were identified as amorphous carbon, originating from the degradation of the C_{60} electron transport layer. This degradation process was found to be decelerated in the presence of a metal interface with the electron transport layer and halted in dry air.

Chapter 1: **Introduction**

Globally we are facing a climate crisis. Since the Industrial Revolution, atmospheric carbon dioxide levels have been steadily rising, hugely due to our unsustainable uses of fossil fuels [1]. In 2024 the global atmospheric carbon dioxide levels reached new record highs, as shown in Figure 1.1 below. These levels of CO_2 are detrimental to our survival on Earth, as the rise of insulating greenhouse gases like CO_2 and CH_4 will continue to cause global temperatures to rise. [2] Without change, this will be catastrophic for all living things in the near future. Urgent global action must be taken in order to combat human-made global warming and climate change.

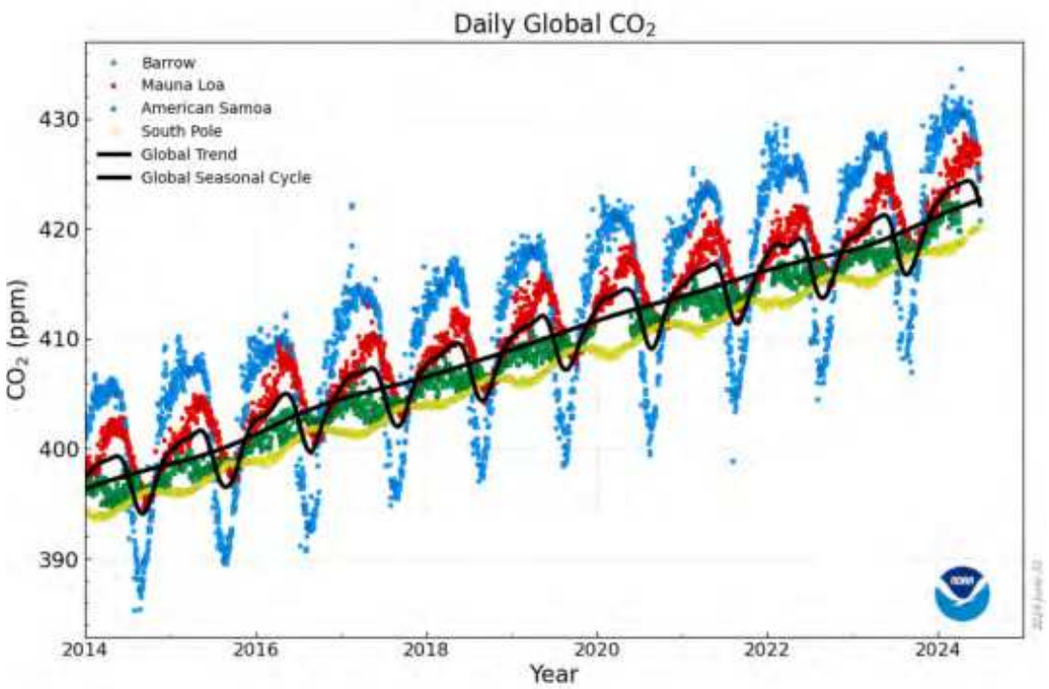


Figure 1.1: The daily averaged CO_2 from four GML Atmospheric Baseline observatories: Barrow, Alaska (in blue), Mauna Loa, Hawaii (in red), American Samoa (in green), and South Pole, Antarctica (in yellow). Adapted from the United States' National Oceanic & Atmospheric Administration's Global Monitoring Laboratory. [2]

Should we want to limit the rise of global temperatures to 1.5°C, as set in the Paris Agreement we must act efficiently and urgently [3]. The continued use of fossil fuels is unsustainable to meet these climate goals, and hinders our ability to meet the world's soaring demand for energy without further damaging the planet [4]. The International Energy Agency (IEA) has said that strong action in the energy sector is needed to drive the reduction in global greenhouse gases, with renewable energies such as wind and solar photovoltaics (PV) taking centre stage. This action includes the need to triple global renewable energy capacity by 2030. [5] This renewable energy capacity growth is predicted to be dominated by the growth of solar PV and wind, with them accounting for up to 96% of the growth in the most ambitious case set by the IEA (see Figure 1.2 below). Solar energy has the benefits of being modular (i.e. can be made as an individual panel, or several can be connected to form arrays [6]) and producible from large scale factory production. These modules can quickly created and deployed. In addition to this, they are

still affordable to both manufacture and buy. It has both large scale industrial applications as well as smaller-scale residential ones. Crystalline silicon PV still holds the majority of the market share for PV technology (approximately 90% in 2018 [7]). However new and more efficient designs and materials are coming to the fore. [8]

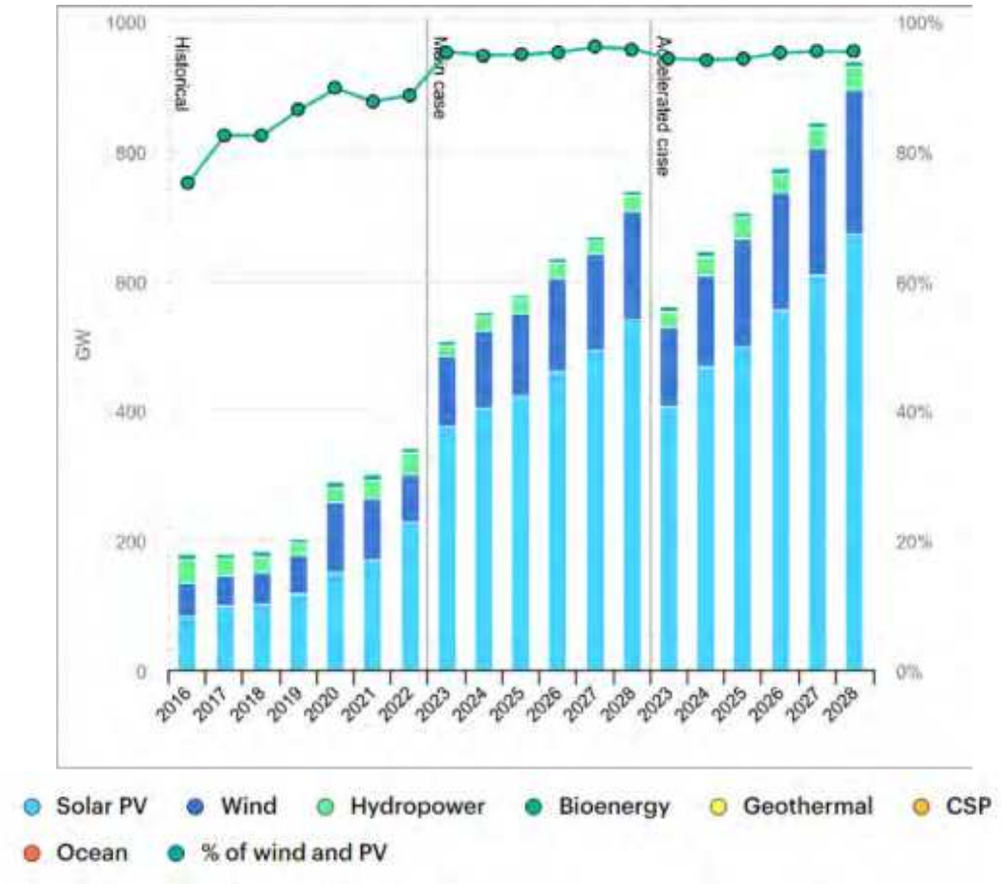


Figure 1.2: The renewable electricity capacity additions by technology and relative proportion, with historical data and predicted regular pace and accelerated pace cases, for the time range 2016-2028. Adapted from the IEA Solar PV interactive chart webpage. [9]

Metal halide perovskite solar cells have been a growing area of interest in solar PV research since they were first introduced in 2009. [10]. Metal halide perovskite materials are very attractive as a solar PV material. This is due to their their superb optoelectronic properties, such as direct bandgaps, high absorption coefficients, low exciton binding energies, and efficient charge transport properties. They can be used to create perovskite solar cell devices that are light-weight, and can be produced through low-cost, low-temperature (usually below 130°C), and roll-to-roll fabrication methods. They can also be applied to flexible substrates meaning more versatile applications for the technology. [11, 12] This potential for relatively cheap upscaling makes them a decent rival to silicon PV, which has inherently high manufacturing costs [7]. These metal halide perovskite solar cells (PSCs) have been shown to have growing potential in their power conversion efficiencies (PCE). [11, 13]

The potential for relatively high-efficiency solar cells that could rival crystalline silicon solar cells compliments the goals set out by the IEA. The theoretical maximum efficiency that can be achieved by a photovoltaic device can be described by the detailed balance limit [14]. In Figure 1.3 below, we see an illustration of the fraction of the detailed balance limit achieved by various record-efficiency cells of different materials. From this we see that single-junction metal halide perovskite record-efficiency cells have almost reached the same efficiency as crystalline silicon. This record has been achieved by a perovskite

device with a bandgap of approximately 1.6 eV. We can also observe that the ideal bandgap to achieve the maximum efficiency possible set by the DB limit is in the range of 1.1 - 1.4 eV. [15] Theoretically, PSC efficiency could be improved by decreasing the bandgap of the perovskite material. Metal halide perovskites have typically been made with lead ions as the metal. It has been seen that by substituting some of these lead ions with tin ions, enables bandgap tuning to lower bandgaps [13]. This can be made possible by taking advantage of the bandgap bowing phenomenon created when alloying lead with tin in the B-site of a perovskite material crystal, which is in the form ABX_3. [16] This is discussed further in Chapter 3.

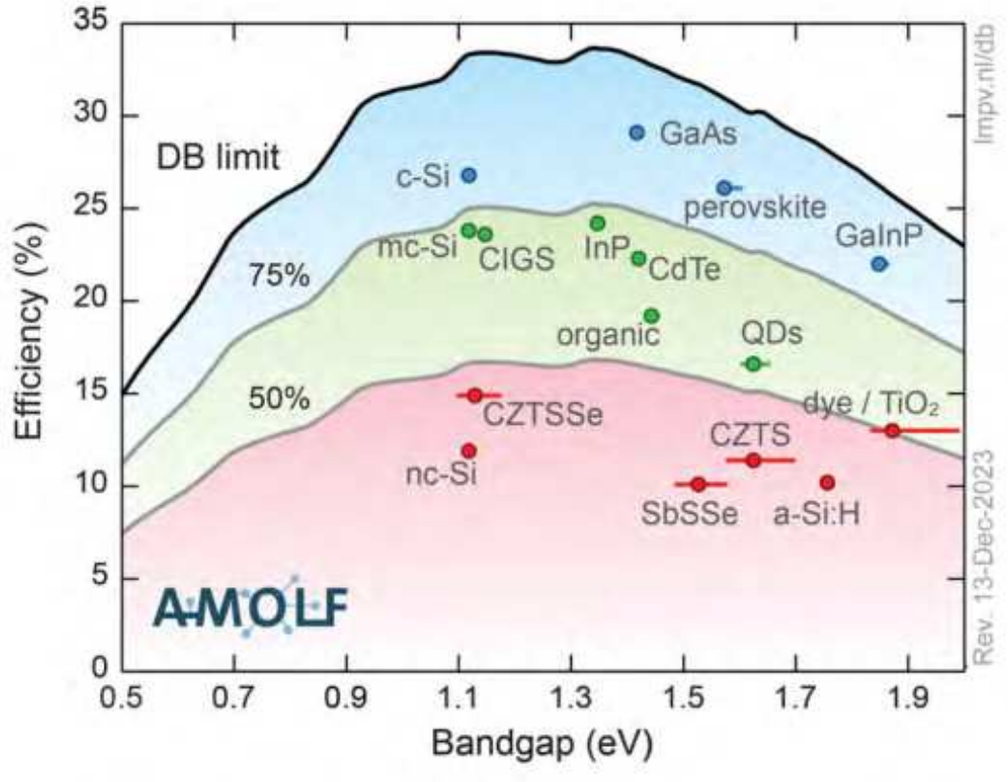


Figure 1.3: Fraction of the detailed balance limit achieved by various record-efficiency cells of different materials. The black line at the top illustrates the detailed balance limit, and the grey lines illustrate 75% and 50% of this limit. Perovskite record efficiency cells can be seen to be placed in the greater than 75% of the detailed balance limit region. Taken from the AMOLF LMPV 'Detailed Balance Charts' webpage. [15]

Solar cells can also be designed in a multi-junction format. This can be a way to bring solar cell efficiency beyond the detailed balance limit set for single-junction perovskites, without adding substantial cost [17]. These configurations allow for more efficient use of solar energy by taking advantage of two materials with different bandgaps to absorb broader portion of the solar spectrum and reduce losses [18]. Perovskites are an attractive material for this configuration due to them being lightweight, and flexible [12]. Metal halide perovskites' tunable bandgap makes them suitable to be considered as the wide bandgap absorber in tandem perovskite/silicon solar cells, and as both the wide and narrow bandgap absorber in all-perovskite tandem cells. The configuration most close to commercialisation is the perovskite/silicon tandem, as a result of its superior efficiency to all-perovskite tandem cells [17, 18]In Figure 1.4 below we see that the record research-cell efficiency held by all-perovskite tandem cells and silicon/perovskite tandem cells are 29.1% and 33.9% respectively. [19]

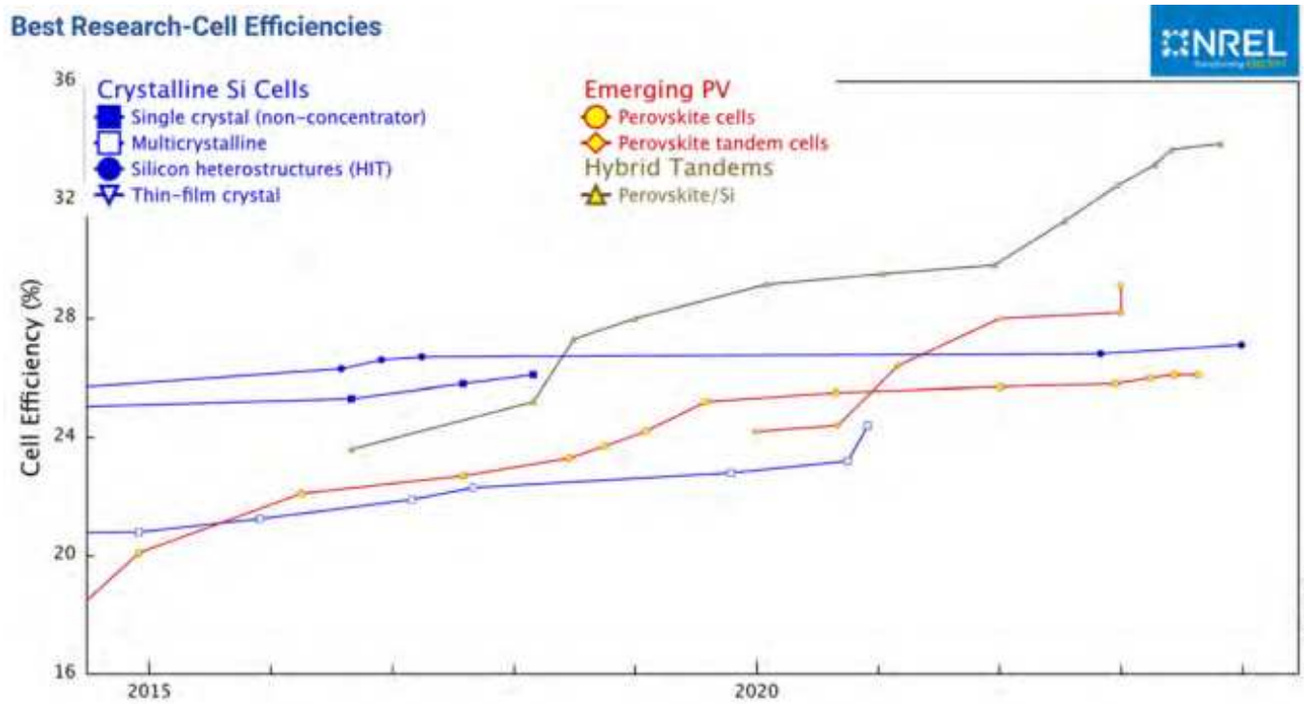


Figure 1.4: The highest confirmed conversion efficiencies for research cells for single junction silicon solar cells, perovskite solar cells, all-perovskite tandem cells, and silicon/perovskite tandem cells. Adapted from the NREL Interactive Best Research-Cell Efficiency Chart webpage. [19]

However, while these efficiencies are already similar to silicon, metal halide perovskites suffer from a number of stability issues, that are detrimental to the commercial viability of perovskite solar cells. This includes degradation and loss of performance through the presence of moisture, oxygen, heat exposure, and light exposure. They are summarised in Figure 1.5 below. [20]

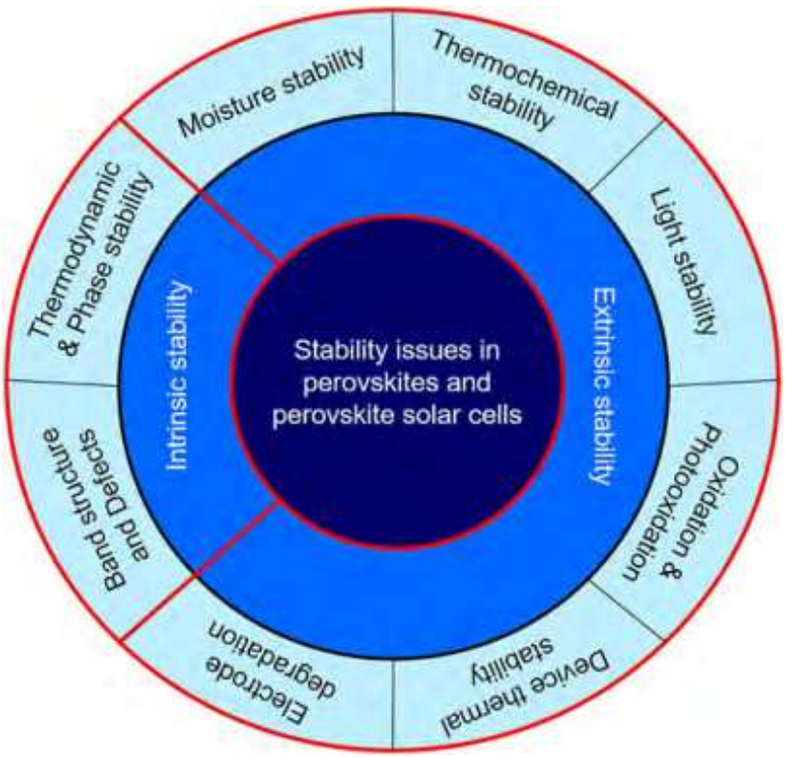


Figure 1.5: Overview of the stability issues seen in perovskites and perovskite solar cells. Taken from Mazumdar et al, *Frontiers in Electronics*, (2021). [20]

In order to compete with other PV technologies they need to have a lifetime that is comparative to silicon, which is approximately 25 years [13, 21]. The International Electrotechnical Commission is the world's leader in setting standards for electric technologies that are available commercially. This means that their tests set minimum stability assessment of any commercialised solar cell [22]. Silicon solar cells already pass these tests [23]. Their tests include ageing PV technologies under exposure to varied humidity, heat, changing climate, and light (including UV radiation). Therefore, perovskite solar cells will need to pass these tests to meet the minimum stability standards of a commercialised solar cell. [13, 24]

In order to pass these tests and be a competitive technology in the industrial market place, the stability issues of perovskites need to be addressed and understood. The contribution of the of interfaces within the PSC between the perovskite and its functional layers to the degradation of the PSC as a whole is still not completely understood [25]. It is imperative to understand how each layer may improve or deplete both the long- and short-term stability of a PSC.

In this thesis, we will examine the influence of the presence of the interface between the electron transport layer and perovskite on the degradation of the samples. The solar cells will be devices using a $Cs_{0.15}FA_{0.85}Pb_{0.5}Sn_{0.5}I_3$ perovskite absorber layer. The specifics of these layers are discussed in Section 3.2. The contribution of the interfaces to the degradation of the PSC will be done by employing a variety of characterisation techniques and fabrication of different samples (see Section 4.1). Through this study, we aim to improve the understanding of the interfaces of lead-tin perovskite solar cells and how it affects perovskite stability over time under different modes of stress.

Chapter 2: **Sustainability**

One of the main motivations for the development of perovskite solar cells is to increase the potential efficiency for photovoltaic technologies. Large-scale, cost efficient technologies will be important in facilitating the transition from fossil fuels to green energy. Multi-terrawatt scale PV module production is expected to begin in order to fulfil this [26]. The IEA has outlined this rapid increase of solar technology as necessary in their pathway to net-zero by 2050 (see Appendix A) [27]. However, if we are to accept this new technology as a green, sustainable material it is not just enough to consider its lack of greenhouse gas emissions during its use. There are many other factors to consider in its life-cycle, such as the mining of raw materials and their abundance, and the impacts on the environment during their operation and end-of-life [28].

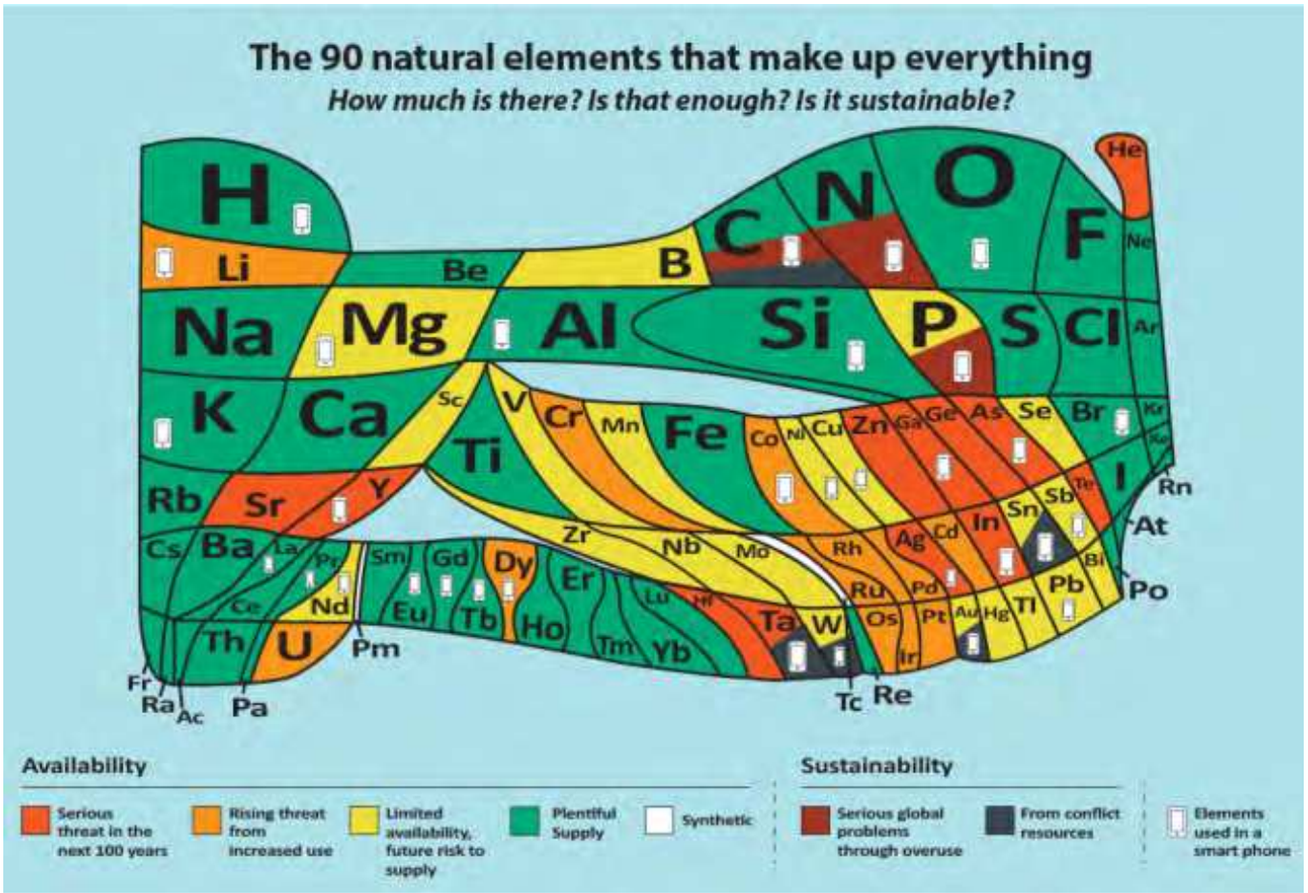


Figure 2.1: The element scarcity and sustainability of materials in the periodic table. Adapted from the European Chemical Society *Element Scarcity* webpage. [29]

In Figure 2.1 above we see an illustration of both the scarcity of each of the elements in the periodic table, as well as an assessment of their sustainability in terms of how they are sourced. Expanding the lifetime of elements, avoiding exhausting resources, and reducing human rights violations are in line with the United Nations Sustainable Development Goals (UN SDGs). Elements are considered to come from 'conflict resources' if they come from mines in countries where wars are fought over the ownership of the mineral rights, where human rights abuses may be taking place. [30–32] We see that a number of these

materials that are considered either to come from endangered sources or from conflict sources overlap with the materials needed for significant upscaling of PSCs below in Figure 2.2. Focusing on the PSC materials used in the duration of this project, we would be looking at indium (In), lead (Pb), tin (Sn), Iodine (I), Caesium (Cs), formamidinium (FA), Copper (Cu), Silver (Ag), buckminsterfullerene (C_{60}), bathocuproine (BCP), PEDOT:PSS, DMSO, and DMF.

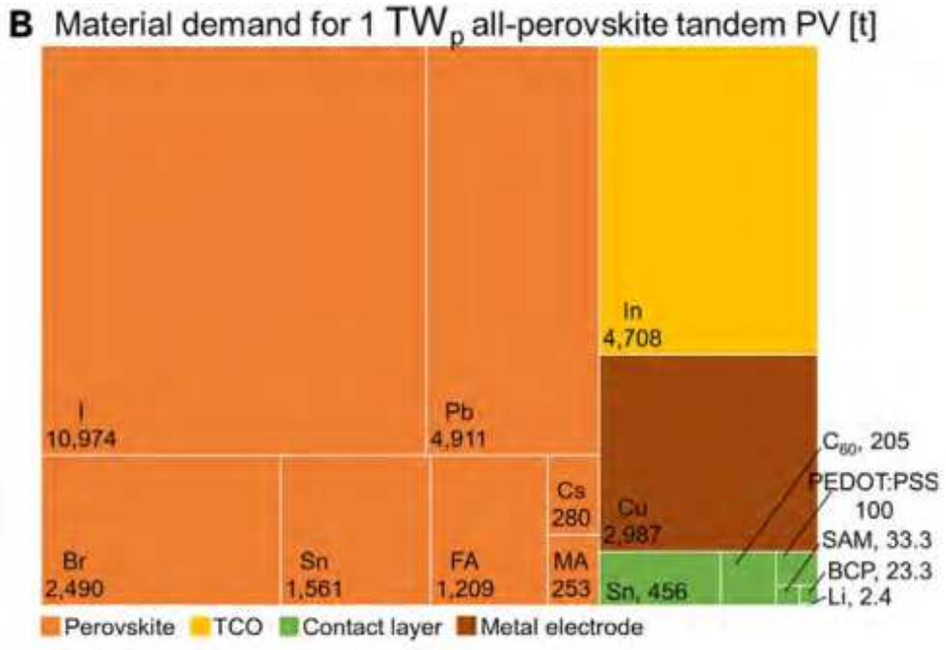


Figure 2.2: The material demand (in metric tonnes) needed to produce 1 TW_p of all-perovskite tandem device structure. Adapted from Wagner et al, *Joule* (2024) [26].

There are a number of materials in this project's perovskite that are suitable for scale up for perovskite PV. In Wagner et al's paper published in 2024, they outline their suitability and readiness for scale-up. They found that formamidinium production exceeds the demand by a factor of 10^5. They also found that in terms of the charge transport layer materials, which would be C_{60}, BCP, and PEDOT:PSS, the C_{60}, and PEDOT:PSS are suitable for large-scale production of PSCs. However, not all materials are as readily available for upscaling. The other charge transport material, BCP, will be highly challenging to upscale as no concept for industrial production has been formulated yet. In terms of the electrode materials, copper and silver have different levels of abundance and availability for ethical sourcing. One tenth of the silver production in the world is claimed by the photovoltaics industry. Silver is considered a material that is within serious threat of running out of supply without suitable recycling infrastructure in the next 100 years [29]. Copper is considered a cheaper and more sustainable alternative. For the transparent conductive electrodes, indium is of particular concern. Indium is considered to have high global supply criticality, and is also considered to be under threat of depletion within the next 100 years [29]. It is highly recommended that substitutions for indium be researched. The solvents used for the perovskite precursor, DMF and DMSO are considered to be in need of further research for upscaling, but will ultimately be plentiful enough in supply for PSC production. This will be helped with the potential research into suitable solvent recycling. [26]

The actual perovskite materials are of varying levels of abundance and sustainability. Caesium is considered abundant, however due to the necessary supply needed for perovskite technologies upscaling, it would be considered in critically low supply for PSC devices. Iodine is considered abudant in supply, although increased production of it will be necessary in order for it to reach the necessary level of supply. Lead is considered to be of limited availability, with potential future risk to its supply, but with currently no threat to its availability for upscaling PSC technologies. Tin is in the same abundance category, but comes with

the added caveat of only a very limited supply available from non-conflict resources [30]. Each of these individual material supply issues will need to be considered. [26, 29].

The risk of resource exhaustion is not the only factor to consider when viewing the overall sustainability of a technology. Looking further into PSCs' life-cycle, we must also consider their potential impact on the environment. Lead toxicity is an ongoing concern of research into PSC technology. Considerable amounts of research have gone into reducing the toxicity of lead-containing perovskites [33]. Lead is considered incredibly toxic, and is classified as a 'substance of very high concern' under the European Union's REACH regulations, as it can be detrimental to human and animal health, and to ecosystems, should it leach into the environment. It is particularly dangerous as it can accumulate over time in soil or groundwater, making it likely to be taken up by plants, and accumulate in their tissues. This can impact plant physiology, nutrient uptake, and crop quality, posing potential risks to food safety and risks to human health if consumed [34]. Children and foetuses are at the most risk for potential health hazards. The potential health risks can include cardiovascular disease, chronic kidney disease, high blood pressure, fertility issues, and adverse pregnancy outcomes. [35, 36] In a study done by Li et al in 2020 on lead uptake in mint plants grown in perovskite-contaminated soil, they found that the lead that can be leached into soil is 10 times more bioavailable than the lead that naturally occurs in soil, resulting in an increased concentration of lead in the entirety of the plant (roots, leaves, and stems) [34]. These concentrations can be seen in Figure 2.3 below.

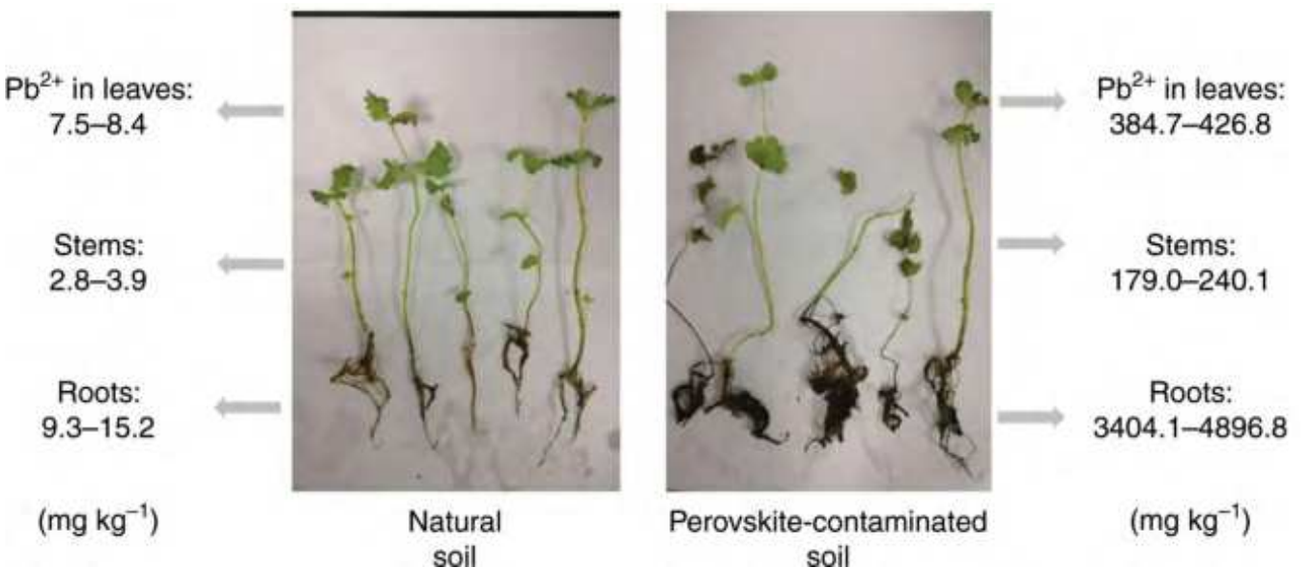


Figure 2.3: Mint plants grown in natural and perovskite-contaminated soil for 20 days. The lead content of each compontent of the plant (root, stem, leaves) are noted. Adapted from Li et al, *Nature Communications* (2020) [34].

Substitution of lead ions with tin ions in perovskite has been a popular mitigation strategy to reduce the toxicity of the perovskite material [37]. However, while metallic tin is much less toxic than metallic lead, its halide salts (which are the form it takes in perovskite materials) are not necessarily much less toxic. There has been a lack of information on the toxicity of SnI_2, and their material data sheets classify both PbI_2 and SnI_2 as accutely toxic under Regulation (EC) No. 1907/2006, although SnI_2 has a lower lethality onset. [28] Further research needs to be done on the potential environmental and health hazards that may be caused by the leaching of lead-based, tin-based, and mixed lead-tin perovskites, as they may all have different reactions and levels of toxicity.

Value Chain Stage	Causation	Health and Safety Risk Type
PSC Design	• Selection of Lead vs. Non-lead Containing Perovskites	N/A
Manufacturing	• Solvent Selection • Machining Processes • Manufacturing Processes	• Inhalation • Skin Contact • Contaminated Effluent
Transportation and Staging	• Improper Packaging or Transportation * • Accident During Transportation *	• Skin Contact
Installation	• Damage During Installation	• Skin Contact
Operation	• Weather Events * • Fire * • Aging *	• Water Contamination • Soil Contamination • Skin Contact • Inhalation
End of Life Management	• Disassembly, Transportation, and Staging * • Landfill or Recycling Processes	• Water Contamination • Soil Contamination • Skin Contact • Inhalation
* Resulting in damage or compromise to module integrity.		

Table 2.1: Potential environmental and human health risk pathways for PSC modules across different value chain stages. Adapted from Rencheck et al, *Solar Energy* (2024) [38].

This leaching of perovskite materials from the PSC modules can occur from a variety of events throughout their life-cycle - from production, to use, to disposal. An outline of these potential risk pathways, and their causes, across the life-cycle of a PSC module, made by Rencheck et al. in their 2024 paper, can be seen in Table 2.1 above [38]. This highlights that ensuring proper protection, handling, and disposal of the solar cells is crucial to avoid environmental damage. Furthermore, this emphasises the importance of improving the stability of perovskite materials under adverse environmental conditions.

Overall, perovskite solar cell technologies have the potential to be a formidable contender in the energy transition from fossil fuels to green, renewable energy. They have been found to be climate-friendly, with the potential to significantly reduce the GHG emissions [39]. However, they still have many factors to be researched in depth and developed in order for PSCs to be scaled up to the appropriate energy capacity, without continuing the cycle of exhausting limited resources, furthering human rights abuses, and straying from a truly sustainable, circular economy. Further research should mainly be focused on suitable replacements for resources in critical supply, maximising the recycling potential of each of the PSC components, and consideration of ethical sourcing of the materials from non-conflict regions. An emphasis on correct handling and encapsulation of the PSCs in order to limit their leaching into the environment during their lifetime and disposal is also of utmost importance.

Chapter 3: **Theory**

3.1 Perovskite Solar Cells

Perovskites are a material with a specific crystal structure that can be described by the formula ABX_3. Metal halide perovskites are a subfamily of this material. Each site in the crystal can be occupied by a different type of ion, illustrated in Figure 3.1 below. The A-site of the perovskite is typically occupied by an organic cation, such as methylammonium ($CH_3NH_3^+$, MA^+) or formamidinium ($HC(NH_2)_2^+$, FA^+) or an alkali metal cation like Caesium (Cs^+). The B -site of the crystal structure is occupied by divalent metal cations, like lead (Pb^+) or tin (Sn^+). The X-site is occupied by halide anions, like iodine (I^-), bromine (Br^-), or chlorine (Cl^-). [40] In the crystal structure, the B-site cations form BX_6 octohedra with the X-site anions, with the A-site cations located at the interstices between these octohedra. [10] The substitution of different ions in the A, B, and X sites allows for facile tuning of the composition of the perovskite crystal. [41].

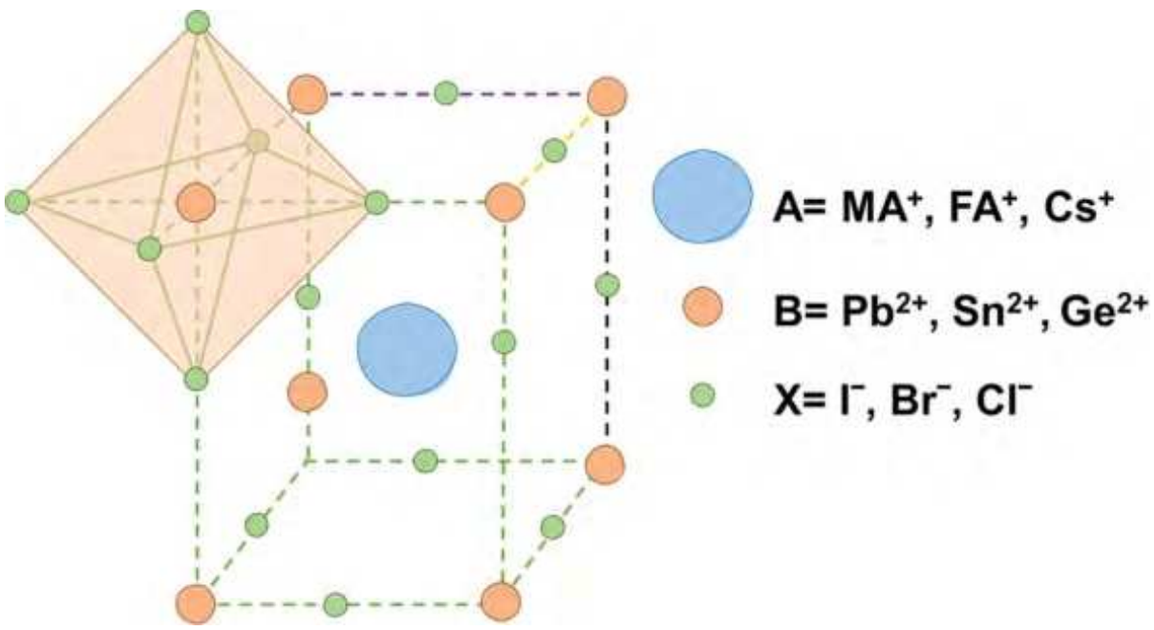


Figure 3.1: Diagram of the crystalline structure of metal-halide perovskite. A denotes the monovalent cations, B denotes divalent cations, and X denotes the halide anions. Taken from Aktas et al, *Communications Materials*, (2022). [40]

Perovskites have incredibly attractive properties that make them ideal candidates for photovoltaic technologies. These include high absorption coefficients, direct bandgaps between ∼1.2 and ∼3.0 eV[42], tunable bandgaps from correct selection of mixed cations and halides to change the perovskite composition, high charge-carrier mobilities, good separation of electrical charges, and high diffusion lengths for holes and electrons. [10] Furthermore, its material properties of being flexible, lightweight, and cheap to manufacture make them an even more ideal material for PV. The exact properties of the perovskite material will depend on its composition [12]. This influences the electronic band structure, charge transport, traps and defects in the material [42].

3.1.1 Lead-Tin Perovskites

Lead is the most commonly used B-site cation, however it has been found that substitution of lead for other cations can change the properties of the perovskite material. This can be done by the substitution of the B-site ions that have different outermost orbitals and are within the same periodic group as lead (such as tin). This enables bandgap tuning to lower bandgaps. [13]

The substitution of tin cations into the B-site of the perovskite crystal lattice of metal halide perovskites can be said to be straight-forward as tin is one of few metal species that are capable of occupying this site. [43]. The choice to use a mixed lead-tin perovskite can be dual-pronged. Firstly, as discussed in Chapter 1, the bandgap of mixed-metal lead-tin halide perovskites can be tuned to be between 1.2 and 1.5 eV, depending on its exact composition (see Figure 3.2 below) [16]. This is a lower bandgap than its pure-lead counterpart can achieve. This makes them suitable as the wide-bandgap absorber layer in high efficiency single-junction solar cells and silicon/perovskite tandem cells. It is also suitable as both the narrow- and wide-bandgap in all-perovskite double- or multi-junction tandem solar cells. In this double-junction tandem configuration with silicon, they have the potential to reach a theoretical maximum of ~45% [44]. Secondly, as was discussed in Chapter 2 there have been concerns about the toxicity of lead ions. So, alongside the benefit of lead-tin perovskites having lower bandgap energies, it has been argued that the reduction in lead content of the material has been considered a benefit of substituting tin for lead cations in the perovskite material. [45]

There have been many studies into how the crystal structure and Pb and Sn ions interact resulting in the phenomenon of bandgap-bowing [13]. As we can see in Figure 3.2 below, at an intermediate tin content, there are attractive near-infrared bandgaps. We see that for a variety of mixed lead-tin perovskite compositions, that the bandgap reaches a minimum of approximately 1.2 eV for a mole fraction range of tin between 0.5 and 0.75. After the tin content is increased below 0.5 or above 0.75, the bandgap begins to increase. [16, 46]

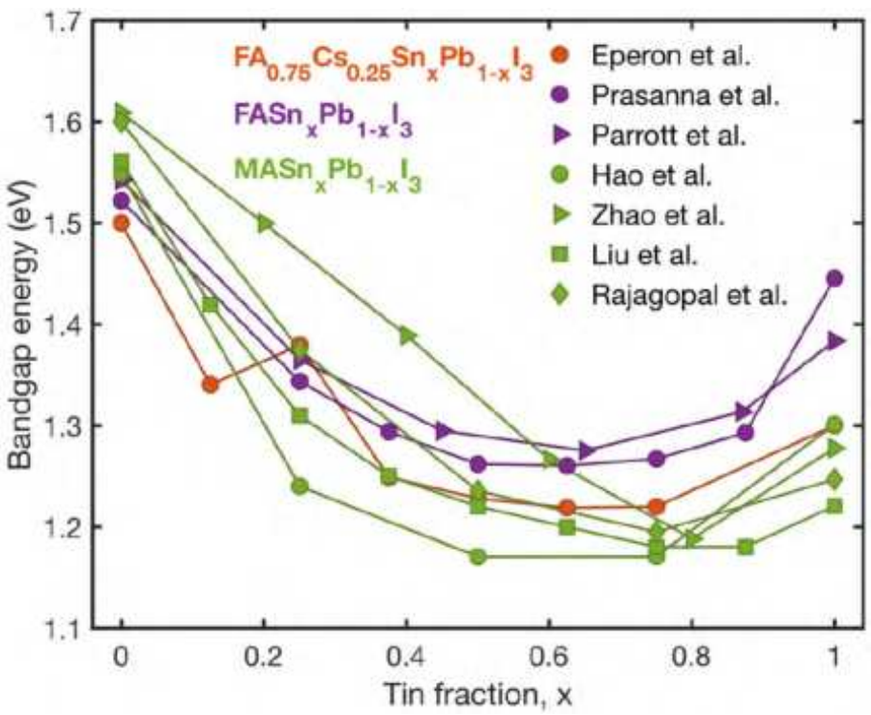


Figure 3.2: Values of the bandgap energy at room-temperature over a range of different literature studies for a variety of different mixed lead-tin perovskite compositions with different A-site cations (MA, FA, or FA/Cs). Adapted from Savill et al, *ACS Energy Lett.*, (2021) [16]

.

Mixed lead-tin perovskites are still not a magic bullet. The downside of this composition is that tin-rich stoichiometries suffer more from stability issues than their pure-lead counterparts. This is discussed further in Section 3.3 below. Therefore, to balance the increased instability of the perovskite material from the introduction of tin, with its bandgap lowering benefits, we can say that the optimal mole fraction of tin is 0.5 [47].

3.1.2 Double Cation Perovskites

The choice of cation in the A-site of the perovskite crystal lattice can have a large influence on its bandgap and stability. Many studies have been done on which organic cation is more suitable for mixed lead-tin perovskite solar cells, MA^+ or FA^+. It has been found that FA^+ is more stable compared to MA^+, as there is intrinsic instability in perovskite films created with MA based on the molecule's volatile nature and thermal instability even at low temperatures. The reasoning behind FA^+ being more thermally stable than MA^+ is its stronger hydrogen bonding with PbX_6 octohedra and reversible decomposition reaction at 85°C. Alongside being more thermally stable, FA^+ has a more red-shifted bandgap due to its large size in comparison to other organic cations. This is more optimal for a solar cell, as it once again will lower the bandgap. [48, 49]

Furthermore, the partial substitution of a smaller, inorganic cation for these organic cations has been examined. It has been found the result of this A-site engineering is dependent on the composition of the perovskites' B-sites. The radius of Sn^{2+} is smaller than Pb^{2+}. As such, we see that incorporating Caesium into the A-site through substituting it for the larger organic cation (like FA^+), will bring the bandgaps of Sn-rich and pure-Sn compositions to a lower value, which is the opposite effect that it has on pure-Pb perovskites.[49, 50] In terms of their combined effect on the stability of lead-tin perovskite compositions, it has been seen that for perovskites with a mix of Cs and FA^+ in the A-site and 50% tin composition, that they have a superior long-term device stability compared to those that contained MA^+. [43, 51]

3.2 Functional Layers and Interfaces

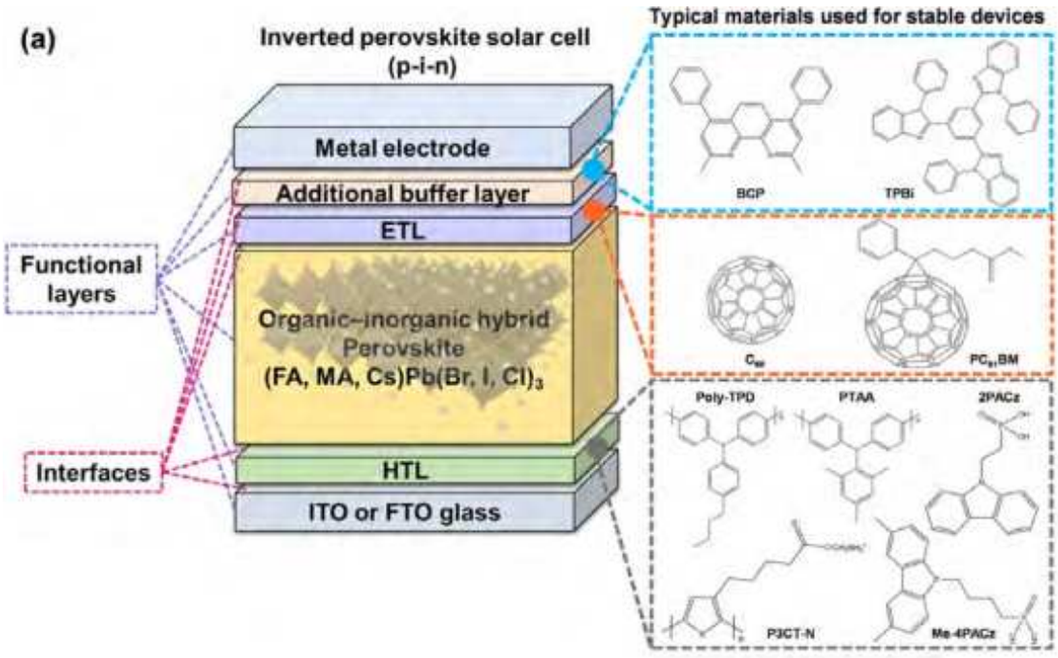


Figure 3.3: Typical planar *p-i-n* solar cell structure. Adapted from Li et al, *Communications Materials*, (2022) [12].

The addition of functional layers to the perovskite absorber layer is necessary for a fully functional and efficient solar cell device. From a variety of studies on PSC performace and stability, it is made clear that the vast majority of high efficiency devices have a "sandwich structure", with multiple functional layers [52]. In Figure 3.3 above we see a planar *p-i-n* solar cell structure. In this typical planar *p-i-n* solar structure, there is a hole transport layer (HTL) deposited onto a transparent conductive oxide (TCO) substrate, to act as the front-end interface. On top of this is the intrinsic absorber layer, which is the perovskite, with an electron transport layer (ETL) and metal electrode deposited on top of it.

Each interface individually, and functioning together within a solar cell as a whole, and the materials chosen for these layers, play an important role in the functionality and performance of PSCs. The structural versatility and customisability of composition of perovskite materials gives rise to a wide range of possibilities for a whole solar cell stack composition. As we can see in Figure 3.4 below, there are many potential options for each layer, and the choice of the correct material for each layer must take not just the composition of perovskite material, but whole stack into account. One must consider that the materials for the HTL and ETL are suitable in a number of ways. This includes if they have a suitable energy level match (to effectively selectively extract charge carriers). They must also have a wide bandgap (to reduce parasitic absorption) and form a good interfacial connection with minimal defects (to reduce non-radiative recombination and charge-accumulation losses). They must also facilitate good charge transfer between the perovskite and the cathode and anode and be relatively transparent, and have suitable deposition for the substrates chosen. When put together as a full-stack device, these layers all-together determine the electronic properties of the device, as well as influencing the stability of the device. [7, 52, 53]

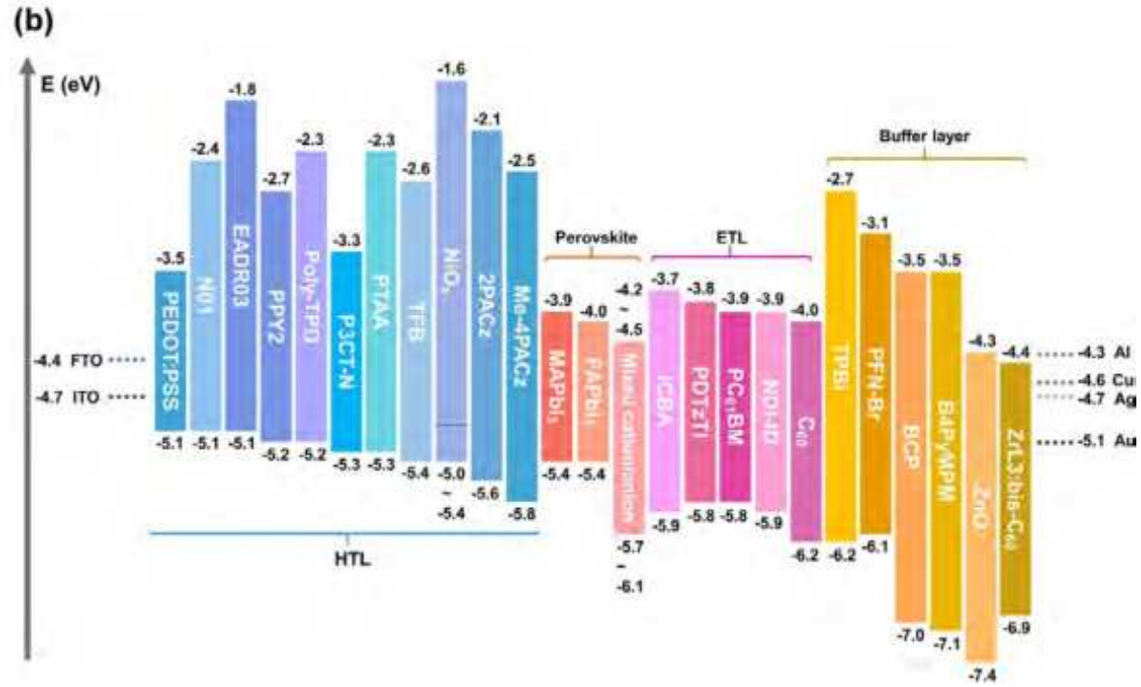


Figure 3.4: Energy level diagram of possible materials for functional layers in high performance inverted perovskite solar cells. Adapted from Li et al, *Communications Materials*, (2022) [12].

Charge transport layers (CTL) (i.e. the electron transport layer and hole transport layer) play an important role in the charge transport, collection, and recombination in PSCs. When the perovskite absorber layer absorbs light, the free charge carriers (electrons in the conduction band (CB) and holes in the valence band (VB)) are generated. These charge carriers drift to the interfaces between the charge transport layers and perovskite. It is during this movement of charge carriers in the perovskite, bulk recombination occurs. The charge carriers pass into the HTL or ETL through the perovskite/CTL interfaces, where interfacial recombination occurs. Interfacial traps can cause first-order recombination. The energy barrier at these interfaces or poor charge transport capability can result in second order recombination at the interface. [7, 52, 54]

3.2.1 Hole Transport Layers

While the HTL/perovskite interface is beyond the scope of this thesis, it still is an important variable in a device stack. The purpose of a HTL is to collect holes generated in the perovskite absorber layer and block the electrons to allow the selective extraction of holes to the external circuit, and reduce carrier recombination [55].

In this project, the HTL used was PEDOT:PSS, deposited onto a pre-patterned ITO and glass substrate (see Sections 4.4 - 4.5.1). PEDOT:PSS has been a popular choice for Pb-Sn perovskites as the HTL due to it being commercially available as a suspension, simple deposition methods, and ability to provide a smooth and hydrophilic surface for perovskite deposition. PEDOT:PSS unfortunately does still contribute to device instability and degradation due to its hygroscopic nature and acidity. [53]

3.2.2 Electron Transport Layers

The purpose of the electron transport layer is to serve as a barrier for holes, and transporting the electrons from the perovskite absorber layer through itself and to the metal electrode. In the *p-i-n* structure, it also operates as a moisture-resistant layer that slows down the degradation of the PSC from exposure to different environmental conditions. Furthermore, the electron transfer at the ETL/perovskite interface is faster than at the HTL/perovskite interface. This means that the choice of the correct material for the ETL is of utmpost importance, in order to optimise the PSCs' performance. [55]

There are pros and cons to using fullerene based electron transport layers (such as C_{60} and PCBM). They have been used in some of the highest performing silicon/perovskite and perovskite/perovskite tandem solar cells [56]. Fullerene materials are commonly used due to their high electron mobility, energy level matching capabilities, and defect passivation of electron-rich defects in the perovskite. Their defect passivation abilities can be attributed to their Lewis acidity. They appear to effectively suppress hysteresis and improve the perfomance efficiencies of perovskite solar cells. [12, 53]

However, despite their part in achieving high efficiencies, using fullerene materials as a sole electron transport layer has been shown to not necessarily be optimal. An ohmic contact between the ETL and its attached electrode is what allows the current flow through this part of the solar cell [55]. C_{60} has an inability to form ohmic contact with the metal electrode layer, thus decreasing charge transfer. Therefore a buffer layer is necessary. Bathocuproine (BCP) is a typically used buffer layer in the ETL. Using BCP as a fullerene buffer layer in the ETL has been shown to yield the highest power conversion efficiency and longest operational stability. These buffer layers unfortunately do oxidise or absorb moisture when exposed to ambient air which can cause degradation of the solar cell, reducing the series resistance, fill factor, and short circuit current density. [12, 53] For the electron transport layer in the stacks made during the project, C_{60} and BCP were used.

3.2.3 Metal Contacts

Making the correct decision on which metal to use for the back-contact metal electrode is very important for PSCs' stability and perfomance. Commonly used materials are gold (Au), silver (Ag), and copper (Cu) [53]. The resistivity of the electrodes can have an effect on the series resistance and fill factor, which will overall affect the power conversion efficiency (PCE) of the PSC devices [55].

The consideration of cost typically excludes gold, and the goal of avoiding at-risk materials (see Chapter 2) can be a factor to exclude silver. However, silver is typically used in *p-i-n* structure PSCs as it has a suitable work function to match the conduction band of ETL materials. Copper is a cheap and abundant material that is electrically suitable for Pb-Sn perovskites and C_{60}. As well as this, it is corrosion resistant and has been shown to give long-term stability even without encapsulation. [55] This makes it a very suitable material for the stacks used in this study.

3.3 Degradation Mechanisms and Factors

3.3.1 Perovskite Degradation

As mentioned in our previous chapters, perovskites are fundamentally unstable under a variety of different environmental stresses. As a material, they have inherent material challenges. They suffer from both intrinsic and extrinsic stability issues. Perovskite films have a non-negligible amount of defects, with defect density scaling with thin film area. As a material, they are prone to lattice distortion and unfavourable ion migration, resulting in easily activated degradation pathways. The components of the ABX_3 crystal each can be volatile and easily react to the presence of moisture, heat, oxygen, or light radiation. [7, 20]

Furthermore, mixed lead-tin perovskites are more unstable than their pure-lead counterparts. This is due to the difference in the stability of the B-site cations. The use of tin presents a challenge for the fabrication of the perovskite film, due to the the Lewis acidity of the Sn^{2+} cation resulting in too fast crystallisation. In comparison to pure-lead perovskites, lead, in its 2+ state, is considered to be its most stable, preferred state. However, this is not the case for tin in its 2+ state. Sn^{2+} is easily oxidised to its Sn^{4+} state, resulting in Sn^{2+} vacancies that act as recombination sites leading to p-doping in the material. This facile oxidation from its 2+ to its 4+ state has been attributed to the fact that the electrons in the s-orbitals of Sn^{2+} ions experiencing a weaker Coulombic attraction to their nucleus than Pb^{2+} ions. This rapid oxidation of the tin ion results in degradation of the optoelectronic properties of the perovskite. [13, 16, 47, 57]

However, examining just tin and lead degradation methods individually is not enough. The exact mechanisms of how tin-lead perovskites degrade are not fully understood yet. While there are clearly outlined degradation pathways for pure-tin and pure-lead based perovskites, it has been shown that the specifics of the mechanism of degradation for mixed metal perovskites are different from their single metal counterparts. The Sn:Pb ratio is an important factor, as it results in a slower rate of oxidation of the perovskite components, as a result of there being less tin present in the material to oxidise. It has been found that the oxidation pathway is a thermally activated one, and that mixed lead-tin compounds will oxidise at substantially higher temperatures than their equivalent pure-tin compositions. [47, 57] It has been outlined that the most likely and prominent defect states are tin interstitials and iodide vacancies [58]. Exposure to ambient air has been shown to be the most detrimental stress factor in terms of the formation of Sn^{4+}, and that lead-tin perovskites do not undergo significant Sn^{2+} oxidation during combined heat and light stressing if they are well-protected from air exposure [59]. In their 2017 paper, Leijtens et al [47] proposed that the alloying of lead and tin in the perovskite lattice can prevent the typical formation of SnI_4 that comes from pure-tin perovskites, and as such the potential two-step oxidation pathways of $FASn_{0.5}Pb_{0.5}I_3$ to be:

$$\begin{aligned} \text{(A) } & 4FASn_{0.5}Pb_{0.5}I_3 + O_2 \Rightarrow 4FAI + SnI_4 + SnO_2 + 2PbI_2 \\ \text{(B) } & + O_2 \Rightarrow 4FAI + SnI_4 + SnO_2 + +2PbO + 2I_2 \end{aligned} \tag{3.1}$$

$$\begin{aligned} \text{(A) } & 4FASn_{0.5}Pb_{0.5}I_3 + O_2 \Rightarrow 2FAI + SnO_2 + 2PbI_2 + I_2 \\ \text{(B) } & + O_2 \Rightarrow 2FAI + SnO_2 + 2PbO + 2I_2 \end{aligned} \tag{3.2}$$

Beyond the additional stability provided by the alloying tin with lead in perovskites, there is also the tactic of the addition of certain additives to the perovskite precursor to pre-emptively combat the degradation associated with tin in the fabrication process. These additives can prevent the oxidation of the tin ions from their 2+ to the 4+ state, potentially reduce Sn^{4+} back to the more optoelectronically optimal Sn^{2+},

limit the fast crystallisation associated with the presence of the Sn^{2+}, and successfully passivate defects. These include the addition of SnF_2 to reduce the oxidation of Sn^{2+} to Sn^{4+} and metallic tin to reduce the Sn^{4+} that already exist in the precursor solution to Sn^{2+}. [57]

As discussed previously, engineering of the A-site is a common tactic to increase the short- and long-term stability of perovskite materials. Investigations into methylammonium (MA)-free perovskite solar cells has been an area of interest for researchers focused on the stability of PSCs. The presence of MA in PSCs have been identified as another factor in device photostability and thermal stability. [60] Instead, a mixture of Caesium (Cs) and formamidinium (FA) in the A-site have been shown to be one of the most stable compositions under heat and light [57].

3.3.2 Interfacial Degradation

Interfaces have a large influence on the stability and performance of perovskite solar cells. Degradation occurs at the interfaces through multiple degradation mechanisms, that can be unique to the materials chosen. As discussed above, their energy band alignment within the stack, and introduction of interfacial defects can have this influence on the performance of the devices. [61, 62] This can include inhibition of charge generation and collection, increased recombination sites, and other appearances of defects [57]. In this project, C_{60} and BCP are the charge transport layer materials being investigated, and copper and silver for the metal contact materials.

C_{60} and BCP

It has been shown that the highest performing *p-i-n* solar cells have been limited by non-radiative recombination that has been induced by the electron transport layer interface with the perovskite absorber layer [56]. This issue has been further examined and has been shown that for fullerene materials, like C_{60}, that used as the ETL in mixed lead-tin PSCs cause energy losses that are dominated by this non-radiative recombination at the perovskite/ETL interface [57]. As well as influencing these energy losses, fullerenes themselves are not inherently stable materials. It has been shown that fullerenes show poor morphological stability, as a result of their relatively weak intermolecular interactions [63, 64]. They also have been shown to be unstable under exposure to UV radiation and high temperatures, which may cause dimerisation and agglomeration of the fullerene materials due to breaking of the fullerene cage [57]. C_{60} in particular has been theorised to introduce deep trap states when put into direct contact with the perovskite absorber layer, resulting in severe interfacial losses [64]. These instabilities in fullerene materials under stress conditions can be detrimental to the overall PSC device stability. Passivation of these defects can be made possible through addition of another layer.

BCP is commonly used alongside C_{60} to improve the ohmic contact with the metal electrode layer. Unfortunately BCP is also not a completely stable material. It has been seen that BCP deposited onto C_{60} can show fast degradation under high temperatures [65]. It has been observed that the BCP can have imperfect film formation with inhomogeneity in its morphology, as a result of it being easily recrystallised [66].

Passivation

Since the direct contact of C_{60} with perovskite has been seen to introduce deep trap states, resulting in large energy losses and recombination at this interface, it has been popular to use an interlayer to mitigate this. By improving these interfacial contacts, the energy losses and recombination at the ETL

can be improved, increasing the efficiency and stability of the PSC devices. For *p-i-n* solar cells, an n-type surface on top of the perovskite layer could reduce traps, reduce the hole-density at the perovskite surface, and improve the charge transfer at this interface. A popular choice for this n-type surface is a small diammonium cation, EDA^{2+}. [64, 67]

For the purposes of this project, the EDAI and IM/CF_3BOM passivating layers (see Sections 4.4 and 4.4), will be examined as a part of the perovskite/ETL interface rather than an interface of their own.

Copper and Silver Electrodes

Metals like copper (Cu) and silver (Ag) are popular choices for tin-lead perovskite solar cells due to their high conductivity and suitable work function [57]. However, these materials are also not stable and can contribute to overall device instability. In particular, when paired with a C_{60} and BCP ETL, it is known that metal contact materials will mix and interact with this layer. This can cause the metals to come into contact with the perovskite, which can facilitate the metal interacting with perovskite ions, drastically effecting the performance of the PSC. This can occur when BCP mixes with the C_{60} instead of conformally coating it, allowing the metal to mix into the ETL. This can result in parasitic absorption from the metal electrode, resulting in optical losses. [45] Copper electrodes have also been shown to undergo oxidation as a result of this direct contact with the perovskite layer [55].

This metal migration into the ETL or perovskite layer and corrosion mechanisms have been seen to occur at temperatures $\geq 70^{\circ}$C [57]. The PSCs being operational increases these mechanisms. Metals like Cu and Ag can readily diffuse into the perovskite absorber layer when activated by heat or light exposure during their use as a solar cell, which can lead to the formation of insulating metal halide species and defect states, both at the now mixed perovskite/ETL/metal electrode interfaces, or in the perovskite bulk [65].

Some degradation can be mitigated by the addition of a metal electrode. It has been shown that in fully encapsulated PSC devices, a copper electrode can operate as a built in desicant for oxygen and water, increasing the lifetime of the device by reducing degradation of the perovskite layer [68].

Chapter 4: **Fabrication**

4.1 Sample Outline

The full stack devices were made on a 1.5 × 1.5 cm pre-patterned ITO and glass substrate (see Appendix B for the pattern). The hole transport layer is PEDOT:PSS, the Electron transport layer is C_{60} and BCP. The metal electrode is copper or silver. See Figure 4.1 below.

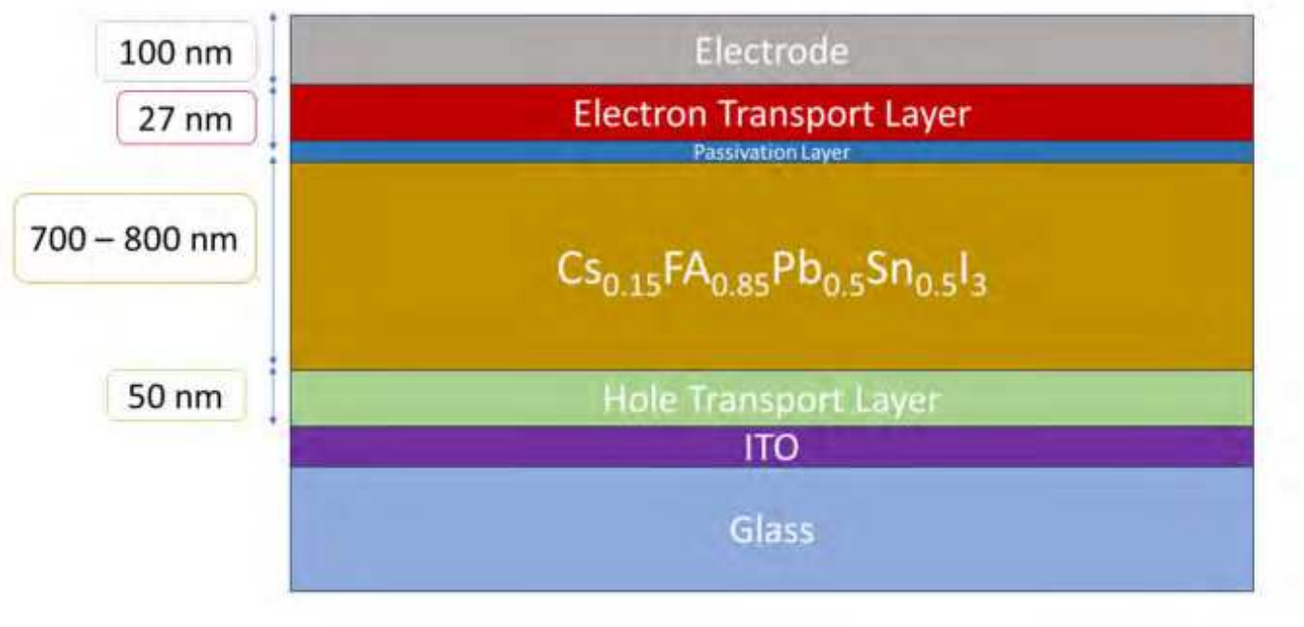
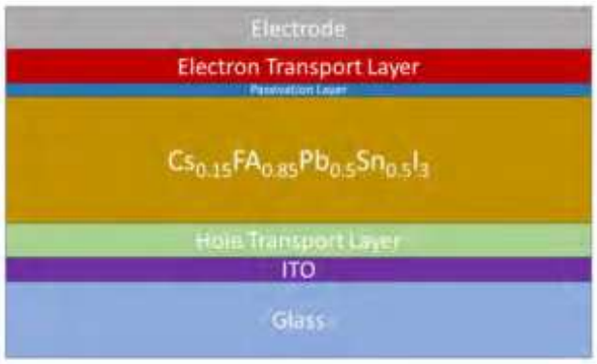
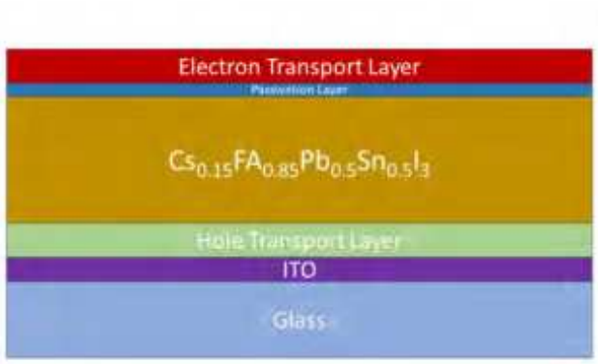


Figure 4.1: Illustration of the estimated thickness of each layer in the full stack PSC device.

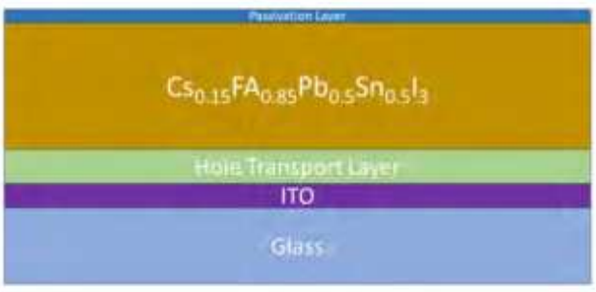
In order to examine the degradation of each layer at the perovskite/ETL/electrode interfaces, a variety of samples must be made. Firstly, a full stack device with all of the layers to examine the contribution of all of the interfaces as well as to act as a control. Secondly, a sample with all layers except for the metal back-contact electrodes in order to examine their interfaces' contribution to the degradation of the solar cell. Finally, a sample without both the metal electrode and the electron transport layer, in order to examine the contribution of the electron transport layers' interfaces. See Figure 4.2 below.



(a) Full Stack



(b) Partial stack, no metal electrodes



(c) Partial stack, no metal electrodes or ETL

Figure 4.2: Overview of the different stacks that were fabricated.

4.2 Substrate Cleaning

The pre-patterned ITO on glass substrates were labelled on their glass-side with a diamond scribe, careful to not scratch through to the ITO pattern. See Appendix B for pattern.

They were then cleaned by carefully scrubbing with a toothbrush and a small amount of Hellmanex Cleaning Concentrate in deionised water. They were then rinsed gently with deionised water.

Once in the holder they were sonicated in a beaker of deionised water, then acetone, then IPA, each for 15 minutes.

Once the sonication was complete, they were dried thoroughly with a nitrogen gun before being placed into a UV-Ozone cleaner for 15-30 minutes, ITO side facing upwards.

4.3 Materials

The perovskite precursor starting materials used were Formamidinium Iodide (FAI, GreatCell Solar Materials, anhydrous, 99.99 %), Caesium Iodide (CsI, Sigma Aldrich, anhydrous, beads, 99.999 %), Lead (II) Iodide (PbI_2, Sigma Aldrich, 99.999 %), Tin (II) Iodide (SnI_2, Sigma Aldrich, anhydrous, beads, -10

mesh, 99.99 %) and Tin (II) Fluoride (SnF_2, Sigma Aldrich, 99 %).

The passivation layer materials used were Ethane-1,2,Diammonium Iodide (EDAI, Sigma Aldrich), imidazole hydrochloride (IM, Sigma Aldrich, 98 %), and 4-(trifluoromethyl)benzamidoxime (CF_3BOM, Sigma Aldrich, 97%).

The solvents used were Dimethyl Sulfoxide (DMSO, Sigma Aldrich, anhydrous, 99.9 %), N,N-dimethylformamide (DMF, VWR International BV, 99.9 %), Anisole (Sigma Aldrich, anydrous, 99.7 %), Toluene (VWR International BV, anhydrous, 99.8 %) and 2-propanol (IPA, VWR Chemicals, anhydrous, 99.7 %).

For the hole transport layer, Poly(3,4- ethylenedioxythiophene):poly(styrene sulfonate) in aqueous solution (PEDOT:PSS, AI 4083, 1:6, Osilla) was diluted with methanol (Biosolve BV, anhydrous, 99.8 %).

For the electron transport layer, Fullerene-C_{60} (Sigma Aldrich, 99.9 %) and bathocuproine (BCP, TCI, 99 %) were used.

For the metal electrodes, copper pellets (Cu, Kurt J. Lesker, 1/8" $\times$ 1/8", 99.999 %) and silver pellets (Ag, Kurt J. Lesker, 1/8" $\times$ 1/8", 99.99 %) were used.

4.4 Precursor Recipes

All precursors were prepared in the inert nitrogen environment of the glovebox, except for the PEDOT:PSS solution which was prepared in ambient air in a downflow cabinet. Each precursor was stirred for a minimum of 4 hours before being filtered.

Perovskite Precursor

Molarity of precursor (mol/L)		2	
Total volume of DMF:DMSO (mL)		1.5	
DMF (mL)		1.125	
DMSO (mL)		0.375	
Precursors	**Molecular mass (mg/mmol)**	**Number of moles (mmol)**	**Mass (mg)**
FAI	171.97	2.55	438.5
CsI	259.81	0.45	116.9
PbI_2	461.01	1.5	691.5
SnI_2	372.52	1.5	558.8
SnF_2	156.69	0.15	23.5
Total weight			**1829.2**
Metallic Sn			**1.8**

Table 4.1: Composition of 1.5 mL of a 2 mol/L precursor perovskite solution.

Precursor materials were weighed to within $\pm$ 0.5 mg. The amounts needed adapted from the precursor recipe from Dey et al, *Energy Environ. Sci.* (2024) [69, 70]. The amount of metallic Sn added is calculated using the total weight of the powders. The SnF_2 is 10 mol% of the SnI_2. These two chemicals are used as an additive to prevent oxidation of the precursor solution (see Section 3.3.1).

EDAI Precursor

The solution was prepared by weighing 2 mg of EDAI into a vial and diluted with 2 mL of IPA and 2 mL of toluene (or any 1mg:1mL:1mL ratio). Recipe adapted from Hu et al, *Energy Environ. Sci.*, (2022) [71].

Imidazolium and CF_3BOM Precursor

This precursor solution for an alternate passivating layer was created by dissolving 2 mg of IM and 1 mg of CF_3BOM salts into 1 mL of IPA. Recipe adapted from Li et al, *ACS Energy Lett.*, (2024). [72]

PEDOT:PSS

Refridgerated PEDOT:PSS was used. In a downflow cabinet in ambient air, 750 μl of PEDOT:PSS was diluted with 2.25 mL of methanol (or any volume with a 1:3 ratio) into a small vial. The vial was then wrapped in parafilm and tape to reduce chances of spilling and placed in the sonicating bath and sonicated for 30 minutes.

4.5 Spincoating

Spincoating was done onto the cleaned substrates. The perovskite and passivating layer precursors were both spincoated manually and using the SCIPRIOS spincoating robot.

4.5.1 Manual Spincoating

In a downflow cabinet, the sonicated PEDOT:PSS solution was filtered using a 0.45 μm PVDF filter into a fresh vial. Then 120 μL of the solution was deposited using a pipette onto the ITO-side of the surface of the cleaned substrates, ensuring the surface is fully covered. This was then spincoated for 30 seconds at 4000 rpm, with a ramp of 1000 rpm/s. Once finished, the substrates were transferred to a hotplate and annealed at 120°C. All samples were taken off simultaneously, once the last sample has been annealed for 20 minutes and immediately transferred to the glovebox. This is done to avoid moisture from the air being absorbed into the samples as PEDOT:PSS is hygroscopic.

Immediately before spincoating, the perovskite precursor was filtered using a 0.22 μm PTFE filter. Then 80 μL of the perovskite precursor solution was deposited onto the PEDOT:PSS covered substrates and spun for 50 seconds at 5000 rpm with a ramp of 1000 rpm/s. After it had been spinning for 25 second 200 μL of anisole was dropcasted onto the spinning substrate. Once finished, the samples were annealed at 100°C for 10 minutes each.

Once annealed, the passivating layer solution was filtered using a 22 μm PTFE filter.

If using EDAI passivation, the substrate was spun at 4000 rpm for 20 seconds with a ramp of 3 seconds, dropcasting the EDAI solution when the final velocity had been reached. The samples were then annealed at 100°C for 5 minutes.

If using IM/CF_3BOM, the substrate was spun at 4000 rpm for 30 seconds with a ramp of 3 seconds,

dropcasting the IM/CF_3BOM solution only after 3 seconds when the final velocity had been reached. The samples were then annealed at 100°C for 8 minutes.

4.5.2 SCIPRIOS Spincoating Robot

The spincoating procedure for the spincoating robot was very similar to the manual spincoating, but with a few different variables to add, and a slight change in the parameters for the ramp and annealing times and temperatures. The discussion on the annealing time and temperature can be read in Section 6.1.

The spincoating procedures for the perovskite and passivating layer precursors are identical to the above Section 4.5.1, however the SCIPRIOS SpinBot had some restrictions on the ramp time. The minimum ramp speed allowed was 2000 rpm/s.

So, for the perovskite solution, instead of a ramp of 1000 rpm/s up to a maximum velocity of 5000 rpm, this was instead a ramp of 2000 rpm/s up to a maximum velocity of 5000 rpm.

For the passivation solutions, instead of a ramp of 3 seconds up to a maximum velocity of 4000 rpm (which is approximately a ramp of 1333 rpm/s), this was instead a ramp of 2000 rpm/s up to a maximum velocity of 4000 rpm.

The SCIPRIOS spincoating robot allows for more precise control over certain variables in the spincoating procedure. A new variable to be considered in the recipe is the tip height for when the solutions are dropcasted onto the substrates. For all procedures, the tip height was set at 5 mm above the substrate.

The annealing time and temperature was varied over the course of optimising the spincoating procedure (see Section 6.1). The annealing temperature is set between 100°C and 110°C and an addition of 30 seconds to each annealing time.

4.6 Evaporation

The electron transport layer and the metal electrodes were applied using an Åmod thermal evaporator manufactured by Angstrom Engineering under a high vacuum. The samples were placed face down inside a substrate holder.

4.6.1 Electron Transport Layer

For the electron transport layer, 20 nm of C_{60} was evaporated once its radak source was heated to approximately 420°C onto the perovskite surface, at a rate of 0.2 Å/s, with the samples rotating at ∼14 rpm. Then 7 nm of BCP was evaporated onto the C_{60} surface, at a rate of 0.1 Å/s, rotating at ∼14 rpm, using a resistive source.

4.6.2 Metal Contacts

An evaporation mask was used so that the metal electrodes could be deposited onto the surface of the ETL in a certain pattern (see Figure 4.3 below). Then 100 nm of the metal (either copper or silver) was evaporated onto the samples. This was done with no rotation, at a rate of 0.1 Å/s for the first 10 nm of evaporating, then a rate of 1 Å/s for the remainder of the process.

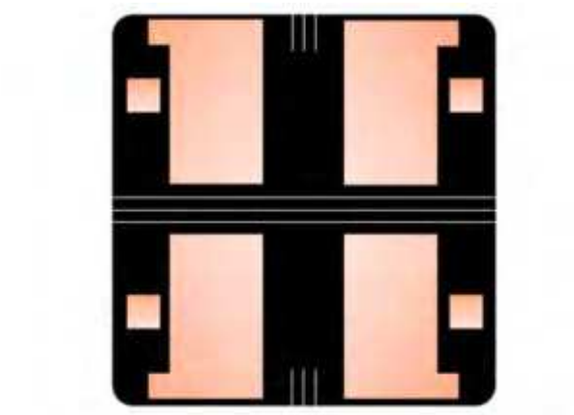


Figure 4.3: Illustration of the metal contact pattern on the full stack devices.

4.7 Device Preparation

Once the fabrication process was complete, the samples needed to be properly prepared for J-V curve measurements (see Section 5.1). This included ensuring both the front and back contacts could be contacted during this procedure. In Figure 4.3 above, we can see 4 small squares of metal in each corner. These were carefully and thoroughly scratched off using a surgical scalpel, to allow contact with the ITO layer. A rubber air duster was used to blow any debris off the surface of the samples.

4.8 Device Storage

All of the samples were stored in opaque black plastic sample boxes to prevent exposure to light. Each box was thoroughly cleaned using IPA and dried with a nitrogen gun to reduce the risk of contamination to the samples.

During degradation experiments (see Section 6.3), the samples were stored in 3 different environments. These environments were in the dry and inert nitrogen environment inside the glovebox, in dry air inside a desiccator, or in ambient air in a sample drawer. The humidity levels between each environment are $<0.1\%$, $\sim 16\%$, and $\sim 72\%$ respectively.

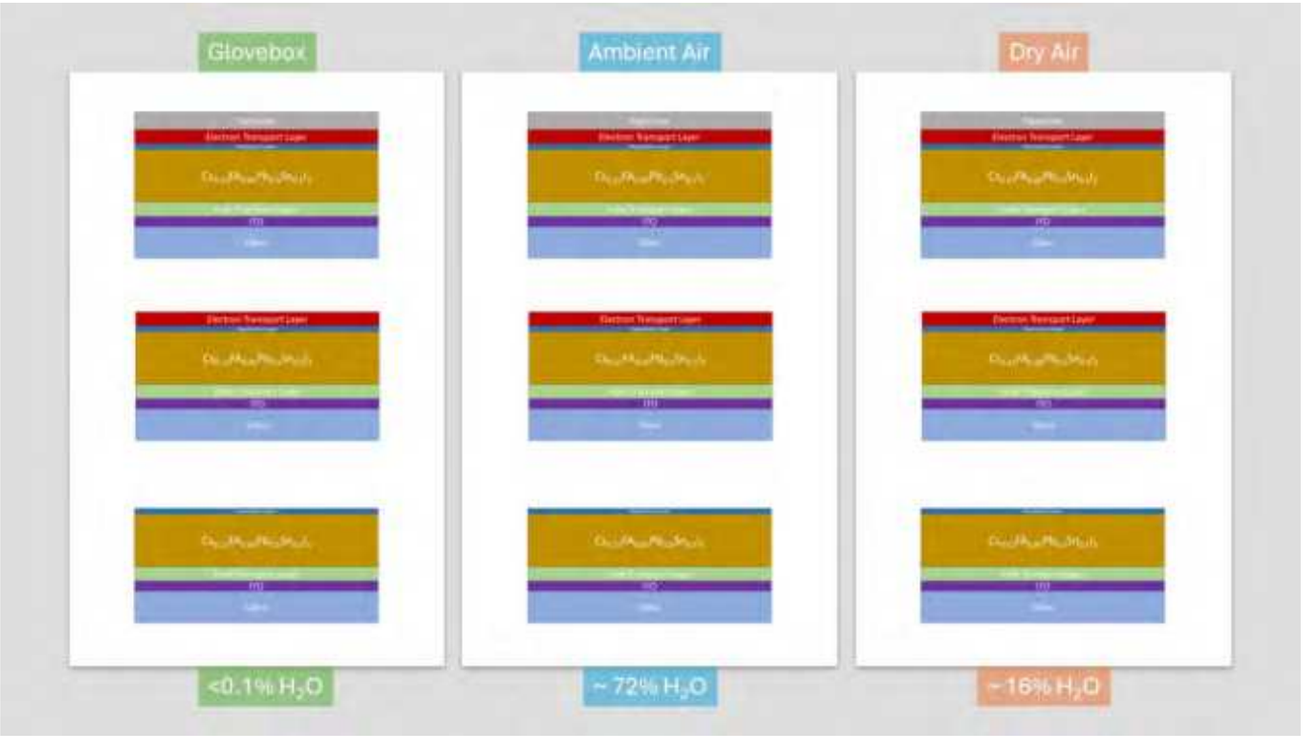


Figure 4.4: Schematic of the storage of devices, and their humidities.

Chapter 5: Methods and Equipment

5.1 J-V Curve Measurement

The current density-voltage (J-V) characterisation of the full-stack solar cells was done using a Keithley 2401 SourceMeter under simulated AM 1.5G one-sun illumination (100 mW cm$^{-}2$) with a solar simulator (PICO G2V), whose intensity was calibrated using a standard silicon solar cell.

The values of the power conversion efficiency of the device were calculated using values obtained from the J-V curve, and using the formula:

$$\eta = \frac{P_{max}}{P_{in}} = \frac{J_{SC} \cdot V_{OC} \cdot FF}{P_{in}} \tag{5.1}$$

where η is the efficiency, P_{max} is the maximum power output per unit area, P_{in} is the incident light power per unit area, J_{SC} is the short-circuit current density, V_{OC} is the open-circuit voltage, and FF is the fill factor.

Light curves were measured individually for each of the 4 pixels within each device (see device layout in Figure 4.3 previously). In order to direct the light to a limited active area when measuring the J-V curves, a metal mask with four 0.065 cm^2 windows was used. The devices were measured from lowest to highest voltage, and subsequently in reverse (-0.3 to 1.2 V, 1.2 to -0.3 V respectively).

In order to measure the dark J-V curves, the same procedure was used except all light was blocked from the sample holder apparatus using aluminium foil covering the lid and mask of the sample holder. The dark curves were taken for each of the 4 pixels within each device.

5.2 Light-Soaking

To investigate the stability of the devices under exposure to light radiation, we performed light-soaking measurements. A python script was ran through Spyder (see Appendix G) to run these light-soaking measurements for 24 hours. For the illumination period, these measurements were taken at 10 minute intervals for the first 2 hours, and then 1 hour intervals for the remaining 10 hours. Then the measurements were taken after 5, 10, 15, and 20 minutes, then after 30, 40, 50, and 60 minutes, and then once an hour for the remaining 11 hours of the recovery period. These 24 hours are so split into a 12 hour application of stress, and then a 12 hour recovery period. The samples were measured inside a light-blocking box to ensure that the devices were only exposed to light during the light-soaking phase and J-V measurements.

All J-V measurements were performed under light in the conditions described in the previous section. These following conditions were only applied to the time period between the J-V measurements.

Light V_{OC}

In order to measure the devices at V_{OC} they were uniformly illuminated by the solar simulator for 12 hours. All 4 pixels were measured for this.

Light J_{SC}

In order to measure the devices at J_{SC}, a bias of 0 V was applied to the devices. The pixels were measured individually for these measurements, with a mask being used to block illumination of the other 3 pixels during light soaking and recovery.

Dark 0V

In order to measure the devices at 0 V no applied bias was needed. The only illumination of the sample was during the J-V curve measurements. All 4 pixels were measured for this.

Dark V_{OC}

An estimated 'V_{OC}' value was used for these measurements of 0.7 V. The only illumination of the sample was during the J-V curve measurements. Pixels were measured individually for these measurements, with a mask being used to block illumination of the other 3 pixels during measurement taking.

5.3 UV/Vis Absorbance

A LAMBDA 750 UV/Vis/NIR spectrophotometer (manufactured by Perkin Elmer) was used to measure absorbance spectra. The sample was placed inside an integrating sphere, with the glass side of the sample facing the beam at a 21° angle. The glass side is used in order to try minimise any reflection effects from the metal contacts on the solar cell.

The sample is measured under ambient pressure, temperature, and atmosphere. The absorbance spectra were measured between 250 nm and 1200 nm at increments of either 0.5 nm or 1.0 nm. The integration time was slowed from 0.1 seconds to 1 second for the wavelength range of 850 nm to 870 nm in order to correct for aberrations in the data caused by the detector changing from the NIR detector to the UV/Vis detector.

5.4 X-Ray Diffraction

X-Ray Diffraction (XRD) was measured at room temperature using a Bruker D2-Phaser diffractometer using Cu Kα radiation, with a wavelength of approximately 1.5418 Å. The 2θ range was from 8° to 50° - 60°, with an increment of 0.024°, and 0.1 second integration time. Each measurement was run for 3 rounds of this range. A 0.6 mm slit and 0.1 mm knife were used for these measurements.

5.5 Scanning Electron Microscopy

The surface of the perovskite samples was examined using Scanning Electron Microscopy (SEM). The surface was imaged using a Thermo Fischer Scientific Verios 460 high-resolution scanning electron microscope. This was typically done at a beam accelerating voltage and current of 5 kV and 100 pA respectively.

5.6 Energy Dispersive X-Ray Spectroscopy

Energy Dispersive X-Ray Spectroscopy (EDX) was carried out on the samples using a Oxford Xmax 80 mm^2 Silicon Drift Detector attached to the Thermo Fischer Scientific Verios 460 scanning electron microscope. The beam current was typically 0.2 nA - 0.4 nA, and between 7 kV - 10 kV. Care was taken to try and minimise the damage done to samples during the EDX analysis, as the aged samples were beam sensitive (see Appendix F).

Chapter 6: Results & Discussion

6.1 SCIPRIOS SpinBot

The quality and performance of perovskite samples can vary between samples within batches and between batches. This highlights an issue with reproducibility in the fabrication process of PSCs, which makes it more difficult to draw solid conclusions of what factors might affect their stability and performance. In order to improve the basis of the results provided and conclusions drawn in this project, firstly the problem of improving the reproducibility between samples was tackled. The fabrication process (see Chapter 4) has many steps, and thus provides a substantial amount of room for human error and variations in the fabrication process. Spincoating encompasses a large amount of the fabrication process, and covers almost the entirety of the portion of the process that is dedicated to the perovskite thin film layer. It was investigated whether automation of the spincoating process for the perovskite thin-film layer and its passivation layer could potentially help improve the reproducibility.

Throughout this project, the samples made for the examination of how the perovskite/ETL/metal contact interfaces may influence the degradation of the lead-tin perovskite solar cells were made with a spincoating robot arm designed by SCIPRIOS (herein referred to as the SpinBot). The SpinBot is designed to manufacture high-efficiency, highly reproducible PSCs with increased stability. This is achieved through the ability to control certain factors in the perovskite fabrication process that would not be possible to keep perfectly constant by manual spincoating methods. This includes the pipette tip height when depositing solutions onto your solar cell substrate, the timing of deposition of solutions and antisolvent quenching, and annealing times. Furthermore, the SpinBot spincoating bowl is designed with a vacuum pump to flush the bowl of potentially harmful atmosphere created by the solvents in the spincoating process, and to pull the nitrogen atmosphere through the bowl in order to optimise the device performance. [73]

6.1.1 Comparison to Manual Spincoating

The first batches that were made in the beginning of this project were done through a manual spincoating method (see Section 4.5.1). As can be seen in Figure 6.1 below, the performance of each solar cell device was widely varied and inconsistent. While the best performing device in this batch had an average PCE of 9.5%, there were other devices within the same batch that were entirely non-functional. As a result, the overall average PCE per device was 1.9%. The outline of the PCE measured for each pixel can be seen in Table 6.1 below.

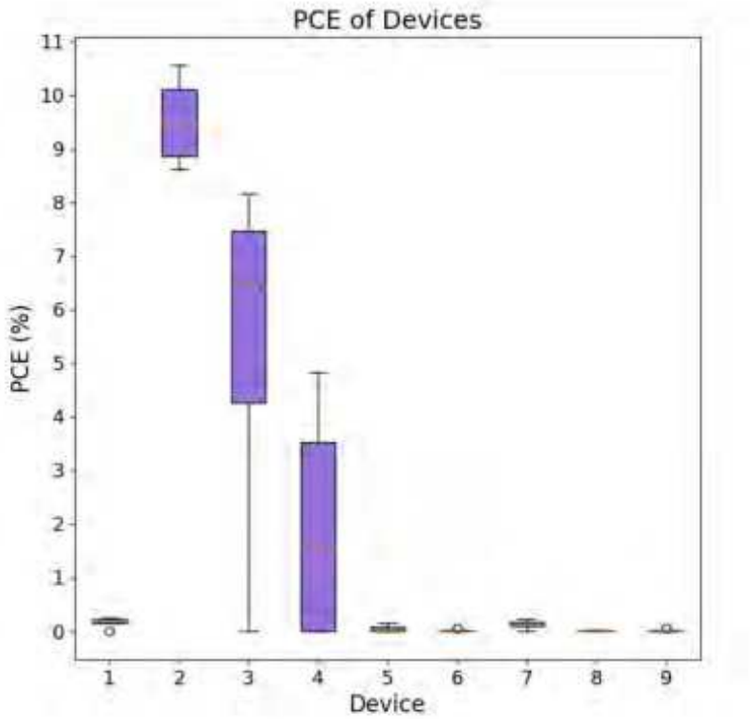


Figure 6.1: Handmade batch of 9 samples made by manual spincoating method. The average device PCE for this batch was 1.9%.

Device No.	PCE Pixel 1 (%)	PCE Pixel 2 (%)	PCE Pixel 3 (%)	PCE Pixel 4 (%)	Average PCE (%)
01	0.22	0.27	0.0	0.2	0.2
02	9.9	10.6	8.9	8.6	9.5
03	8.1	0.0	7.2	5.7	5.3
04	0.0	0.0	3.1	4.8	2.0
05	0.0	0.0	0.0	0.2	0.1
06	0.1	0.0	0.0	0.0	0.0
07	0.2	0.2	0.1	0.0	0.1
08	0.0	0.0	0.0	0.0	0.0
09	0.0	0.0	0.0	0.1	0.0
Mean across all devices (%)					1.9

Table 6.1: Device measurements of PCE across different pixels and average PCE per device as seen in Figure 6.1 above.

Using the same recipe (as identical as was allowed by the SpinBot operating limits, see Section 4.5 for details), the first batch of the SpinBot devices was made. The performance across pixels for each device can be seen in Figure 6.2 below. While there was still obvious variation in performance across pixels and between devices in this batch, the average PCE across pixels per device were 5.0%, 5.9%, and 5.9% respectively. The average PCE across devices was 5.6% in this batch. The outline of the PCE measured for each pixel can be seen in Table 6.2 below.

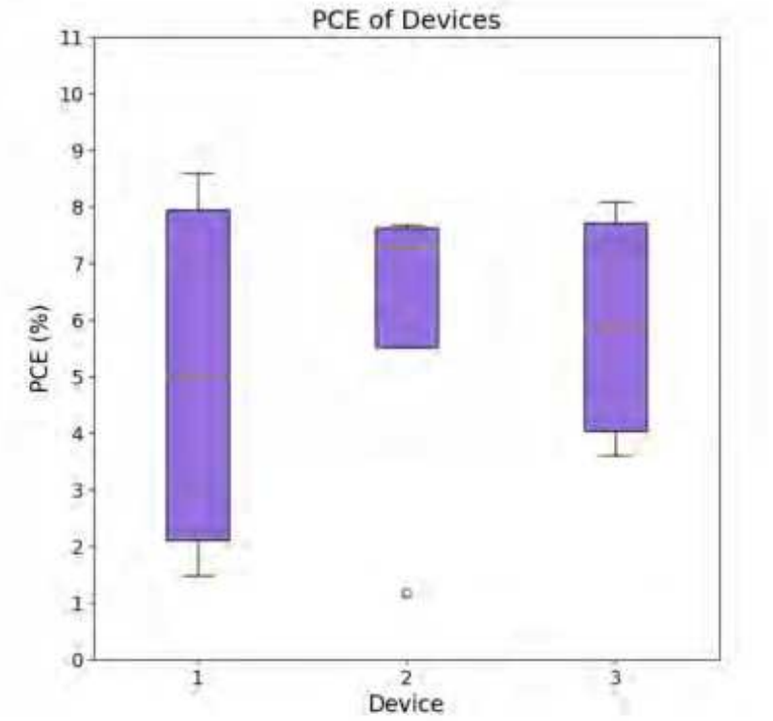


Figure 6.2: Batch of 3 samples made with the SpinBot. The average device PCE for this batch was 5.6%.

Device No.	PCE Pixel 1 (%)	PCE Pixel 2 (%)	PCE Pixel 3 (%)	PCE Pixel 4 (%)	Average PCE (%)
01	7.7	8.6	1.5	2.3	5.0
02	7.7	7.6	7.0	1.2	5.9
03	3.6	8.1	4.2	7.6	5.9
Mean across all devices (%)					5.6

Table 6.2: PCE across different pixels and average PCE per device as seen in Figure 6.2 above.

Over the course of the project, it became clear that the recipe for manual spincoating does not perfectly translate to the SpinBot spincoating method. This can be due to new restraints added to the rotation acceleration of the samples, the introduction of a smaller spincoating bowl, and the introduction of aluminium substrate holders. It was determined that the substrate holders might have been preventing the samples from reaching the correct temperature during the annealing process, so we increased the annealing temperature by 10°C and the annealing time by 30 seconds. This resulted in an increase of average device PCE and consistency between device pixel performance. This can be seen in Figure 6.3 below. The outline of each pixel performance is outlined in Table 6.3 below.

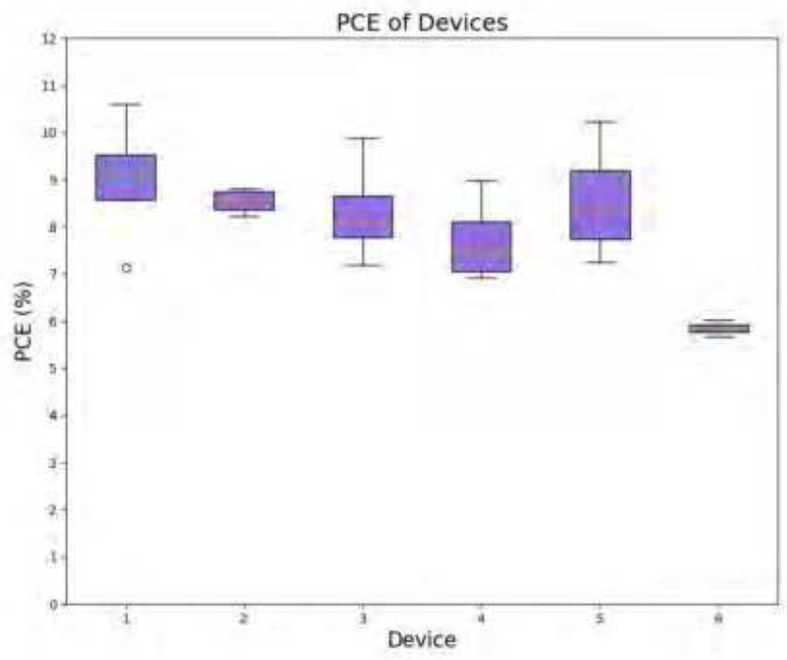


Figure 6.3: Batch of 6 samples made with the SpinBot, with an increased annealing temperature and time. The average device PCE for this batch was 8.0%.

Device No.	PCE Pixel 1 (%)	PCE Pixel 2 (%)	PCE Pixel 3 (%)	PCE Pixel 4 (%)	Average PCE (%)
01	10.6	9.2	9.1	7.1	9.0
02	8.7	8.2	8.8	8.4	8.5
03	9.9	8.2	8.0	7.2	8.3
04	9.0	7.8	6.9	7.1	7.7
05	10.2	8.9	7.2	7.9	8.6
06	5.9	6.0	5.7	5.8	5.8
Mean across all devices (%)					8.0

Table 6.3: PCE across different pixels and average PCE per device as seen in Figure 6.3 above.

6.1.2 Discussion

The purpose of introducing the SpinBot to this project was to increase the reproducibility between samples within a batch and between batches. As discussed in Section 4.1, a variety of different samples are needed in order to examine the contribution of the perovskite/ETL interface, and the perovskite/ETL/metal electrode interface to device degradation. Automating the fabrication of the layers that are included in all 3 of the different types of samples (the perovskite layer and passivation layers) was a logical step to take in trying to increase the reproducibility of these samples.

We can see that, even in the early stages of the study, the first batches were already improved in terms of the consistency of performance between samples by simply switching to the SpinBot. When comparing the batch made by manual spincoating procedure (in Figure 6.1) to one of the first batches made using the SpinBot (in Figure 6.2), we see that the average PCE between devices is improved. This can be determined from observing that all of the devices that were made for the SpinBot batch are functional,

that the spread of performances across devices is narrowed, and the overall average PCE between devices is improved significantly.

A variety of factors were found to affect the translation of the manual spincoating procedure to the SpinBot in order to optimise the quality and performance of the devices. Firstly, the spincoating bowl was smaller and as such the concern about ensuring a good atmosphere for the spincoating was important. As it has been shown that the presence of a clean, nitrogen environment and a vacuum pump will create the best conditions for the fabrication of the solar cell [73], the vacuum pump being on its maximum power was introduced to help pull the nitrogen in the atmosphere into the bowl, while also flushing out any harmful solvent vapour. As well as this, the ramp for the spincoating procedure had to be altered. As discussed in Section 4.5, the SpinBot had restrictions on the ramp time. The minimum ramp speed allowed was 2000 rpm/s, which was faster than what would be used in the manual spincoating procedure. Finally, due to the small size of the substrates we used throughout the project, which were 15 $\times$ 15 mm in size, aluminium substrate holders were designed so that the SpinBot's arm could grip and move the substrates securely around its apparatus. A clear increase in consistency and performance can be seen when these variables are optimised correctly, such as how the annealing temperature and time were changed.

It can be seen that the SpinBot clearly increases the reproducibility of samples within a batch when the recipe has been properly optimised to its specific parameters. The consistency between batches can be said to be somewhat improved. However, direct comparison of the manual spincoating procedure to the automated spincoating procedure is not easily done. While a large section of the fabrication process is made more reproducible, even down to minute details such as the pipette tip height when dispensing precursor solutions and solvents, and highly accurate annealing times, this does not remove all of the other potential variations in the fabrication procedure through the other steps. Human error can be introduced through variations in the weighing of the powders for the precursor solutions, the lack of control over these minutia in the spincoating procedure for the PEDOT:PSS HTL, handling throughout the transfer to the evaporator, and scratching of the ITO contact patches on the samples so that they can be measured. Furthermore, as has already been mentioned, the environment the samples are fabricated in is of utmost importance to the outcome of their performance. The gloveboxes that the batches were made in were shared between many other colleagues and can vary in oxygen and moisture levels depending on how heavily they were being used. Beyond this, thickness in the ETL and metal electrodes were not measured and are instead an estimated thickness.

Overall, the SpinBot can be said to have improved the reproducibility between samples enough that the conclusions drawn between different types of samples, and different batches over the course of the project are valid. This will also be seen in the consistency of the output of their absorbance spectrum, XRD patterns, and consistency of morphology between batches in the following section.

6.2 Light-Soaking

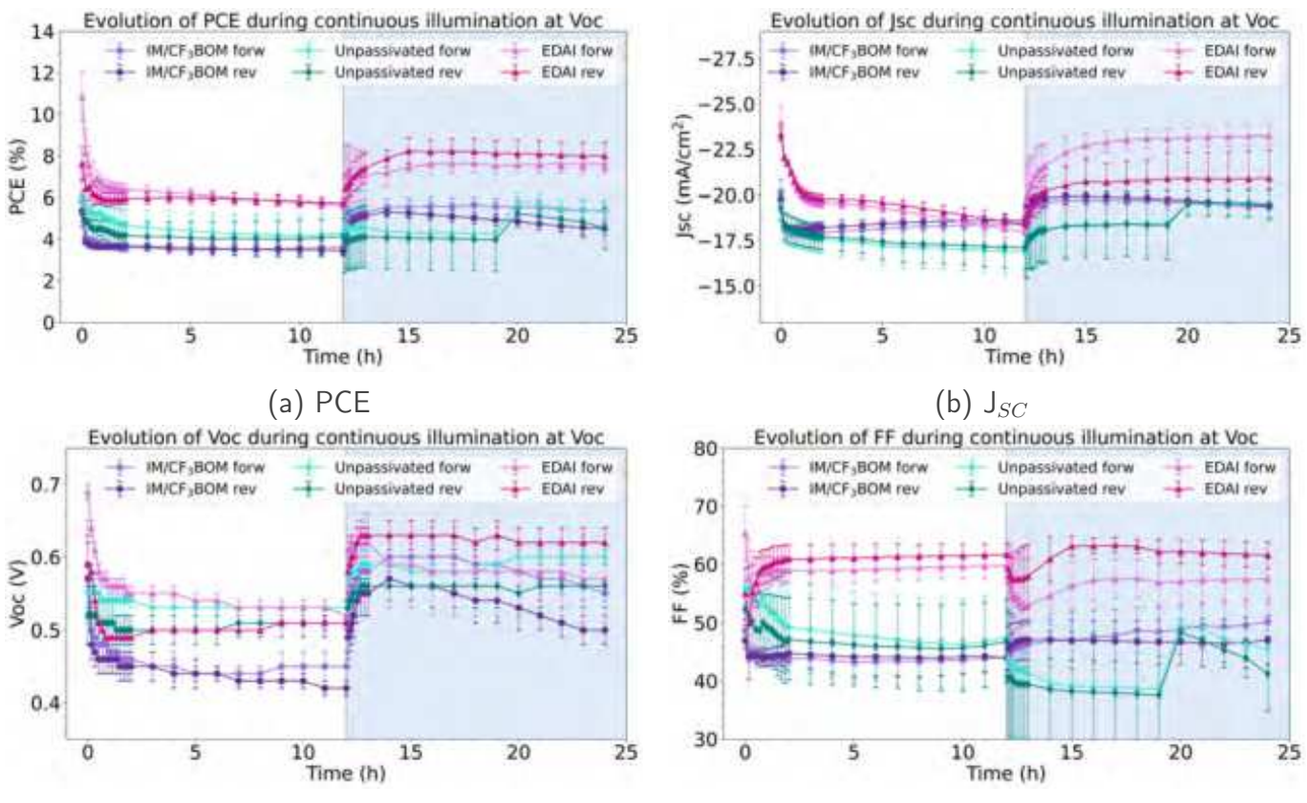
Stability issues in perovskite solar cell devices suffer from instability as a result of light exposure. As we have discussed in Chapter 3, many components of a PSC can undergo degradation as a result of light radiation, particularly UV radiation. We have also seen that this can be exacerbated when the solar cells are in operation. Thus, photodegradation is also a pressing issue in the field of PSC stability research.

Therefore, it was also interesting to examine how the different passivation layers might influence the degradation of these cells under light. This changes the contact at the perovskite/ETL interface. The EDAI sample data were not from this study's batches, and instead were from a batch previously fabricated and measured by Daphne M. Dekker. These samples were made by manual spincoating and not the SpinBot.

Below in Figures 6.4, 6.5, 6.6, and 6.7 we see how each of the J-V characteristics of these devices evolve over the 24-hour measurement period in different environments (12 hours under stress and 12 hours recovery - see Section 5.2 for more detail). We can see that the performance and stability of the solar cell devices can have different effects depending on what type of stress they are exposed to.

6.2.1 Light Measurements

In Figure 6.4 below, we see that when the samples are measured under constant illumination for 12 hours while at V_{OC}, their overall performance drops. We can see that there is a clear sharp drop of the V_{OC} and J_{SC} of all types of devices within the first hour, which causes a sharp drop in their measured PCEs.



(a) PCE (b) J_{SC}

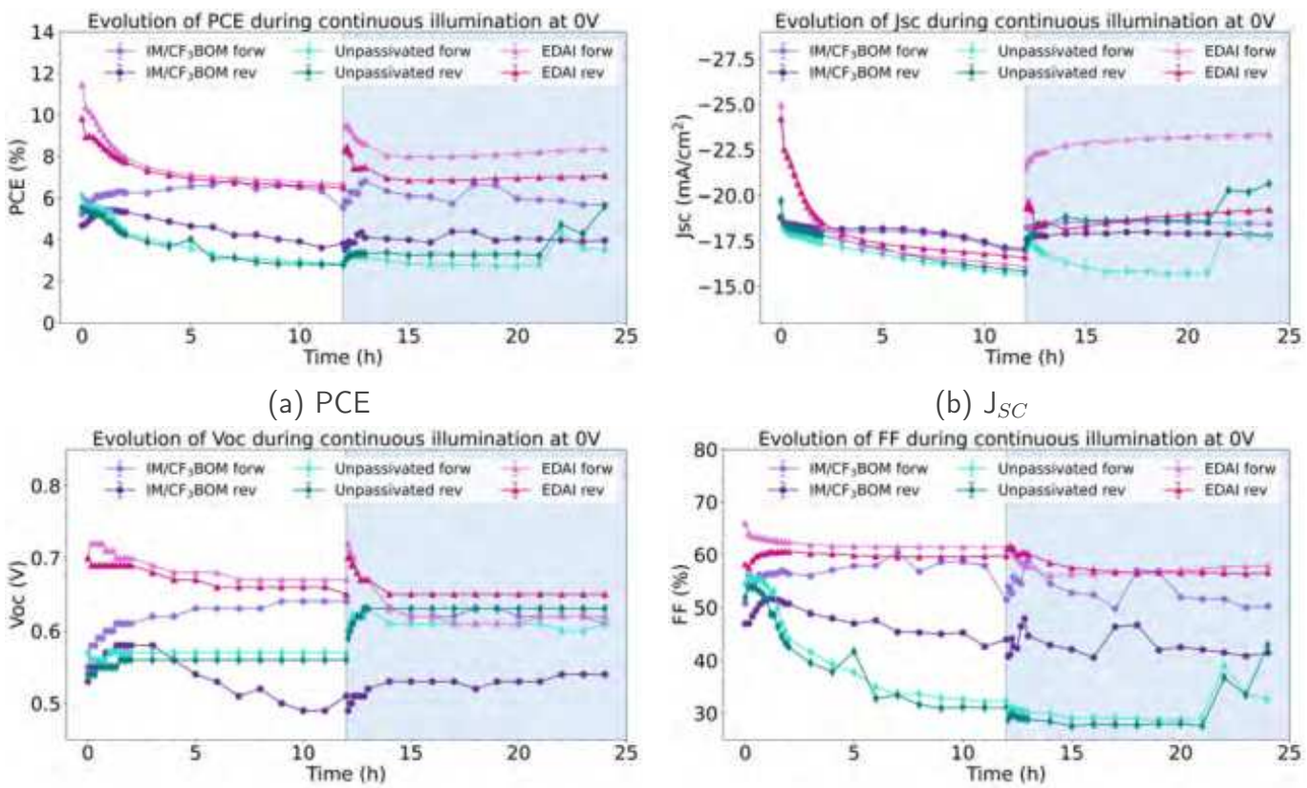
(c) V_{OC} (d) FF

Figure 6.4: J-V characteristics of devices with varying passivation layers for continuous illumination and recovery at V_{OC}.

The J_{SC} of the samples using the EDAI passivation appears to then steadily decrease under illumination, and then quickly recover, albeit with a noticeable difference between the forward and reverse scans. This steady decrease, and then recovery of the J_{SC} is also seen in the devices with no passivation layer, without the difference between the forward and reverse scans during the recovery. Interestingly, the device with the salts passivation layer shows that the J_{SC} begins to increase again slightly and remain relatively steady after the initial sharp drop, although it does not recovery as well during the recovery phase.

The V_{OC} of all the samples appear to generally gave the same trend over the illumination period. The samples using EDAI and with no passivation appear to rapidly recover to an improved V_{OC} after the recovery period begins and stay steady. This is not true for the samples using the salt passivation, which does improve rapidly with the onset of the recovery period, but then once again steadily declines.

The FF of the samples is where we see the most difference overall between each type of sample. The EDAI devices show a sharp decrease and then increase in their FF which then stays steady over the illumination period. We then see a small drop initially after the recovery period begins, but it increases back to its original FF. We see the largest difference between the forward and reverse scans for this device. For the other two types, we do not see this increase once the FF initially drops, although their FFs do also remain steady after the first hour like the EDAI device. However, once the recovery period begins, we see the salts passivated samples slightly increase in their FF and remain relatively steady, showing a small difference between the forward and reverse scans after 17 hours. The unpassivated samples see a sharp decrease once the recovery period begins, and seems to slowly decrease. There is a sudden jump in the FF for this type of device at hour 20, where it then seems to begin decreasing again.



(a) PCE (b) J_{SC}

(c) V_{OC} (d) FF

Figure 6.5: J-V characteristics of devices with varying passivation layers for continuous illumination and recovery at 0 V.

In Figure 6.5 above, we see that when the samples are at 0 V (or at J_{SC}, there is a no longer a sharp initial drop in the PCE, and overall the PCE seems more steady than when at V_{OC}. There is more pronounced

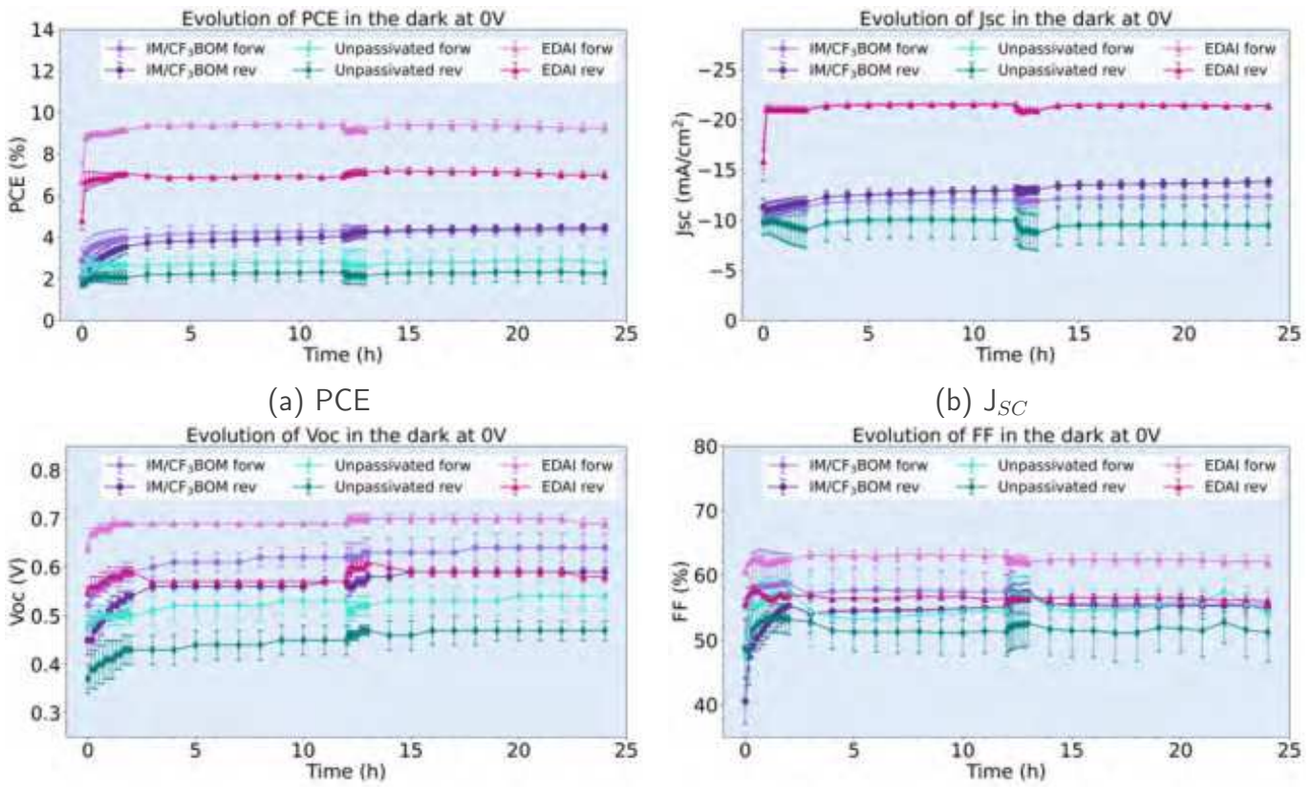
hysteresis in the devices with passivation, particularly the salts passivation.

The J_{SC} of the samples drops to roughly the same level for all samples, most severly decreasing in the EDAI passivation devices. While in the the forward scan, of the EDAI devices improves back to nearly its original value, the reverse scan value remains lowered significantly. The J_{SC} for the salts passivated samples appears to be relatively steady across both the illumination and recovery periods. The unpassivated device shows a steady decrease under illumination, with an improvement during the recovery phase (albeit with a significant difference between the forward and reverse scan values).

The V_{OC} of the samples shows different results for each type of device. The EDAI devices decrease slightly over the illumination period, and continue to decrease and then stay remarkably steady during the recovery period. The salts passivated device show a large difference in the forward and reverse scan values. The forward scan value appears to improve and stay steady, and the reverse scan appears to get worse and then stay somewhat steady at that lower value. The unpassivated device appears to stay very steady in the illumination period, and then improve significantly and rapidly with the onset of the recovery phase, and then stay steady.

The FF of these devices also show different results. The EDAI device generally stay steady, with a slight lowering of the FF at the onset of the recovery period. The salts passivated device also once again show a significant gap between the forward and reverse scan values, which both decrease very slightly over the full measurement period. The unpassivated device shows a fast decrease in FF which does not fully recover, although there is a sharp increase in its values after 21 hours.

6.2.2 Dark Measurements



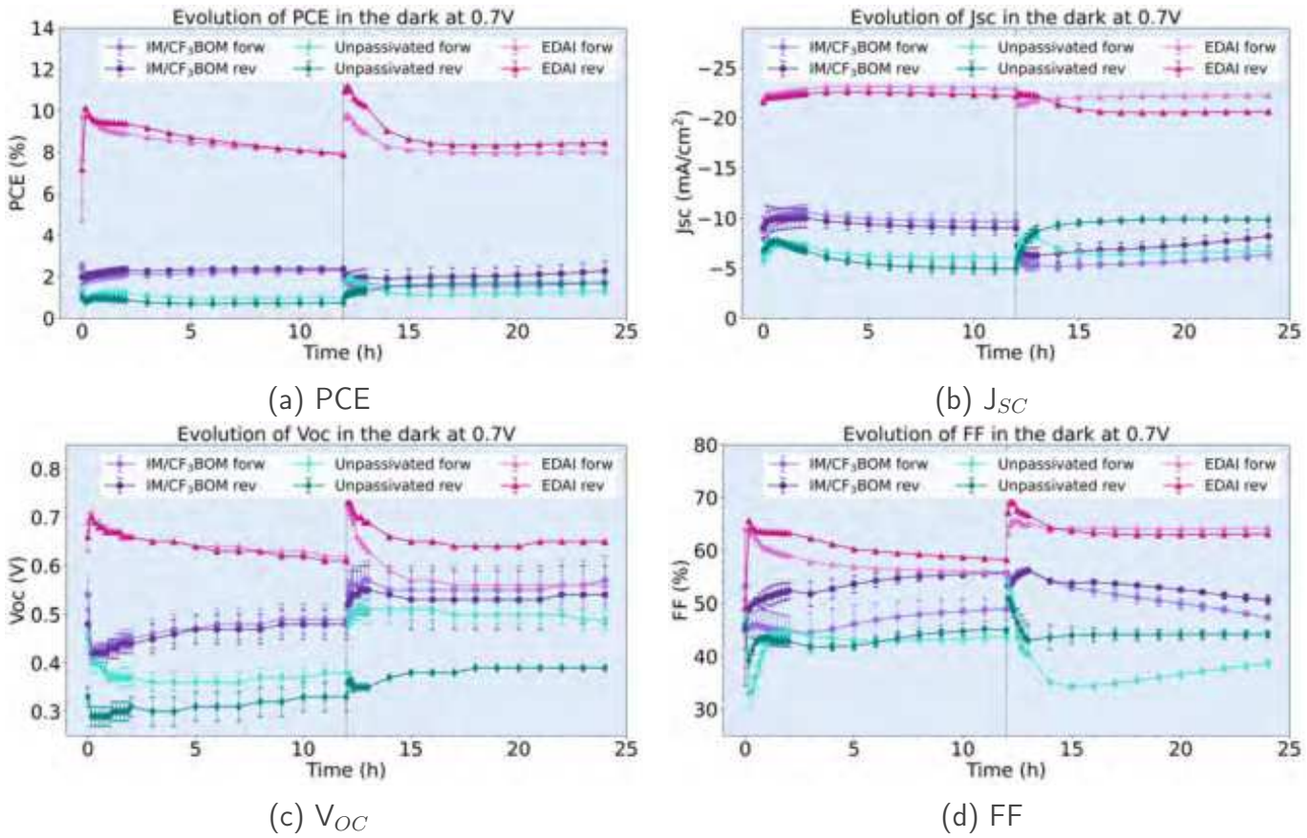
(a) PCE (b) J_{SC}

(c) V_{OC} (d) FF

Figure 6.6: J-V characteristics of devices with varying passivation layers in the dark at 0 V and their recovery.

In Figure 6.6 above, we see the samples when kept in the dark, with no bias applied to them. This will test the devices' stability throughout the measurement period, without any other stresses applied. The overall PCE of each device initially improves within the first 10 minutes, and then all stay steady during the measurement period. Each device maintains the boost in PCE, with the EDAI devices showing the most significant initial increase, and the largest hysteresis.

This is true for the J_{SC}, V_{OC}, and FF for each type of device. The J_{SC} of all devices seems to be very slightly impacted by more frequent measurements (seen at around 12 hours), but this is recovered once less frequent measurements are taken. The V_{OC} is also noted to have continually improved over time for all devices.



(a) PCE (b) J_{SC}

(c) V_{OC} (d) FF

Figure 6.7: J-V characteristics of devices with varying passivation layers in the dark at V_{OC} and their recovery.

In Figure 6.7 above, we see the samples kept in the dark, but during the 'stress' period they are kept at an estimated V_{OC} value of 0.7 V. We see that the PCE is relatively steady for all devices, with the most significant change in the EDAI device. We see that the EDAI device PCE rapidly improves at the beginning of the measurement period and then slowly decreases, but not back to its original value and then has another jump in the PCE with the onset of the recovery, but then settles back down to a steady value similar to what was measured during the stress period. The other two devices remain steady.

The J_{SC} of the EDAI device remains relatively steady over the whole measurement period. The salts passivated device also appears steady during the stress period, with a sharp drop in its values at the beginning of the recovery period, and then slow recovery trending towards its original value (but not completely reaching it). The unpassivated device shows a similar trend as this, but then instead recovers to a higher value during the recovery period.

The V_{OC} of the EDAI device decreases over the measurement period, showing a gap between the forward

and reverse scan values during the recovery period. The salts passivated and unpassivated devices both show an imorovement overall in the measurement period.

The FF of the passivated devices appears relatively steady, with the EDAI device showing an improvement from its original value, and the salts passivated device staying somewhat the same. The unpassivated device shows an improvement during the stress period, followed by a decrease in the forward scan FF during the recovery.

6.2.3 Discussion

Examining the stability of PSC device performances and other J-V characteristics under light exposure and biasing can give insights into certain degradation mechanisms [60]. Furthermore, the C_{60} layer has been shown to also be unstable under UV-radiation. Through variation in the passivation layer, we can change the perovskite/C_{60} interfacial contact, which means we can delve deeper into the interfacial degradation that occurs at this site. By examining this for samples with different passivation layers, we can also glean information on the contribution of the perovskite/ETL interface to their stability. In this section, we see that even simply changing the passivation layer can result in varied performances of the device, and how they degrade under different stress conditions. Only one round of these light-soaking experiments were able to be ran due to time constraints of the project, and as such further repetitions of these measurements would be needed to thoroughly examine the implications of these results.

We see that when the samples are illuminated for 12 hours at V_{OC} the performance of the devices have generally the same behaviour. The most noted difference in the device characteristic trends is in the fill factor values. For the samples illuminated for 12 hours at J_{SC} we see different results. Most notably here we see an increase in hysteresis over the whole measurement period for the salts passivated device, and during the recovery period for the EDAI device. The J_{SC} of the EDAI device also significantly decreases during the illumination period. Investigating the origin of these differences is difficult within the scope of this project as these results were not reproduced and the samples were made using different techniques in different environments.

We see that in general the PCE of the devices is affected by the presence and type of passivation layer used. This shows that interfaces can influence the operational stability of PSC devices. Overall the devices with EDAI outperform the samples with alternate passivation/no passivation - although, once again, since the conditions that these samples were made with were from different operating procedures, different environments, and by different people, only tentative conclusions can be drawn from this.

An important observation can be made from the Dark 0 V measurements. We see that if we do not apply a bias or illumination to the samples, they stay stable (and may even slightly improve in their performance) over the measurement period. This underlines that the changes in stability that we observe for the samples is as a result of the illumination or biases we apply to them. It also highlights that these different stresses will affect the devices differently if they have an altered perovskite/ETL interface.

Further, the devices as a whole are shown to degrade more under light-radiation exposure than when kept under stress in the dark. We do see that the unpassivated devices have some recovery abilities, but at a lower and slower magnitude to their passivated counterparts. Overall, all the samples appear to recover well after light-soaking, implying that these samples are relatively stable under light and can somewhat self-recover.

6.3 Effects of Interfaces on Degradation

We have observed how full stack solar cell devices may degrade in an inert, nitrogen environment under light exposure and electrical biasing. This highlighted the role of how altering interfaces in the device may change the operational stability of the cells. Thus, interfaces in a PSC device will affect the degradation of the PSC over time. In order to investigate how other forms of stress may influence the degradation of PSC devices, and how the ETL and metal contact interfaces might contribute to this, we examined the following.

The role of the ETL and metal contact interfaces, with each other and with the perovskite, in the degradation of the lead-tin perovskite solar cell was investigated by way of preparing 3 different types of samples (see Section 4.1), to examine the degradation of the solar cell as a whole, and how this may change as layers are removed. In order to examine the degradation, J-V curves were measured, the UV-Vis absorbance of the samples was measured, the crystal composition and structure was examined using EDX and XRD measurements, and the morphological changes were examined through SEM imaging.

In order to examine the origins of each type of degradation, three different storage environments were employed. Samples were either stored in the inert, dry nitrogen environment in the glovebox, in dry air, and in ambient air (see Section 4.8). Within each degradation batch, a set of all 3 samples were made using the same perovskite precursor, and all fabricated within the same time period to ensure as much consistency between samples as possible.

6.3.1 Metal Contacts

First we investigate the contribution of all the layers, including the metal contacts. These samples were the full stack devices. For the most part the devices were made with copper as the back contact electrode, however silver was used for one batch.

Ambient air degradation

These samples were made using the SCIPRIOS SpinBot as discussed in the previous section. The performance of the devices was measured when they were freshly made (>18 hours after evaporation), and then 7 days after. In Figure 6.8 below, we see the PCE spread over the 4 pixels for each device. Half of these samples were stored inside the glovebox, and half outside. These samples were made with the initial SpinBot recipe, without the adjusted annealing time and temperature. The outline of the PCEs measured for each pixel and the overall device averages per type of device can be seen Tables 6.4, 6.5, 6.6 and 6.7 below.

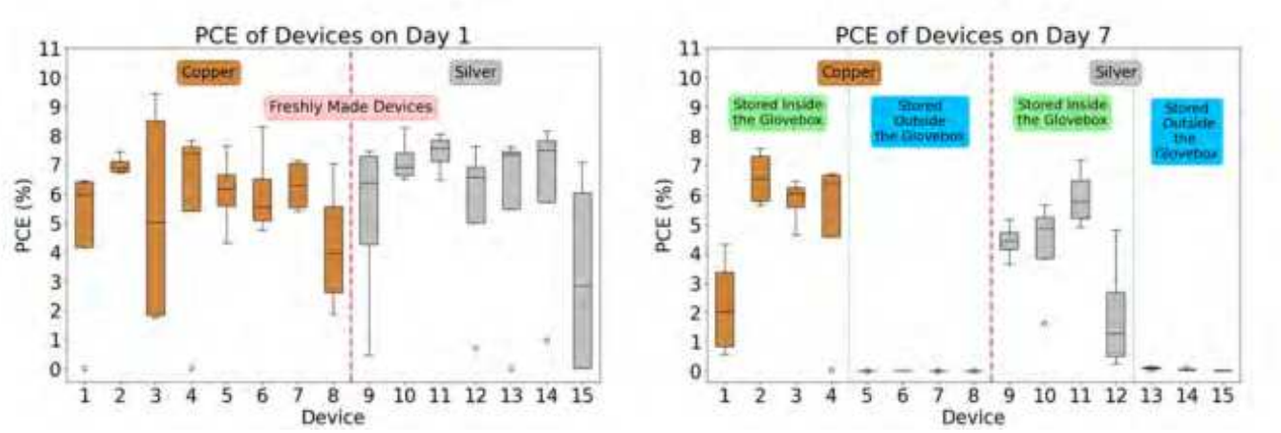


Figure 6.8: Batch of 15 samples made with the SpinBot, made with the original SpinBot procedure. 8 samples were made with copper electrodes, and 7 were made with silver electrodes. Half of each type of device were stored inside the nitrogen environment of the glovebox, and the other half were stored inside a sample drawer.

Copper Devices					
Device No.	**PCE Pixel 1 (%)**	**PCE Pixel 2 (%)**	**PCE Pixel 3 (%)**	**PCE Pixel 4 (%)**	**Average PCE (%)**
01	6.4	0.0	6.4	5.6	4.6
02	7.4	6.7	7.0	6.8	7.0
03	9.4	8.2	1.8	1.9	5.3
04	7.5	0.0	7.8	7.2	5.7
05	7.6	6.3	6.0	4.3	6.1
06	8.3	5.2	5.9	4.8	6.0
07	5.4	5.6	7.2	7.0	4.2
08	2.9	7.0	1.9	5.1	5.6
Mean across all devices (%)					5.6

Table 6.4: Device measurements of PCE across different pixels and average PCE per freshly made devices with copper electrodes seen in Figure 6.8 above.

Silver Devices					
Device No.	**PCE Pixel 1 (%)**	**PCE Pixel 2 (%)**	**PCE Pixel 3 (%)**	**PCE Pixel 4 (%)**	**Average PCE (%)**
09	0.5	5.5	7.2	7.5	5.2
10	8.3	6.7	7.1	6.5	7.2
11	8.0	7.3	6.5	7.8	7.4
12	0.7	6.7	7.6	6.4	5.4
13	0.0	7.4	7.6	7.3	5.6
14	0.9	7.3	8.1	7.7	6.0
15	0.0	0.0	5.7	7.1	3.2
Mean across all devices (%)					5.7

Table 6.5: Device measurements of PCE across different pixels and average PCE per freshly made devices with silver electrodes seen in Figure 6.8 above.

We can see in the graph above, the performance of the copper and the silver appear to be approximately

the same in the fresh batch. Each batch of samples was fabricated in sets of 8. There appears to be a slight trend downwards of performance for devices along the 7th and 8th devices for each batch (device number 8 for the copper, and device number 15 for the silver).

Once left to age for 7 days in the glovebox, we see a small difference in the performance between the different metal electrodes. The copper samples appear to have retained most of their perfomance, whereas the performance of the silver samples has dropped (see Tables 6.6 and 6.7 below).

Copper Devices					
Device No.	**PCE Pixel 1 (%)**	**PCE Pixel 2 (%)**	**PCE Pixel 3 (%)**	**PCE Pixel 4 (%)**	**Average PCE (%)**
1	3.1	4.3	0.6	0.9	2.2
2	7.6	7.2	5.8	5.6	6.6
3	5.9	4.6	6.5	6.2	5.8
4	6.1	0.0	6.7	6.7	4.9
Mean across all devices (%)					4.9

Table 6.6: Device measurements of PCE across different pixels and average PCE per devices with copper electrodes stored in the glovebox for 7 days, as seen in Figure 6.8 below.

Silver Devices					
Device No.	**PCE Pixel 1 (%)**	**PCE Pixel 2 (%)**	**PCE Pixel 3 (%)**	**PCE Pixel 4 (%)**	**Average PCE (%)**
09	5.2	3.7	4.3	4.6	4.4
10	1.6	4.6	5.7	5.1	4.2
11	4.9	5.3	6.3	7.2	5.9
12	0.6	0.3	2.0	4.8	1.9
Mean across all devices (%)					4.1

Table 6.7: Device measurements of PCE across different pixels and average PCE per devices with silver electrodes stored in the glovebox for 7 days, as seen in Figure 6.8 below.

As for the samples that were stored outside the glovebox, in ambient air, their ability to function as a solar cell appears to have been depleted altogether as we can observe in Figure 6.8 above. When examined for other batches, we also saw that this performance drop to 0% can be seen within the first 24 hours of exposure to ambient air.

The samples stored outside the glovebox had their visual appearance changed. The samples became less opaque in appearance, more red in colour, and the samples with the yellow-ish tint caused by the electron transport layer lost this yellow colour (see Appendix C). This change in optical properties was then studied using UV-Vis absorbance measurements. As we can see in Figures 6.9 and 6.10 below, the samples that were stored inside the glovebox experienced almost no change in their absorbance spectrum, for both devices with the copper and the silver electrodes. However, for the samples stored outside the glovebox, we see a shift of the absorption onset into the bluer wavelength range. See Appendix D for more frequently measured absorbance spectra.

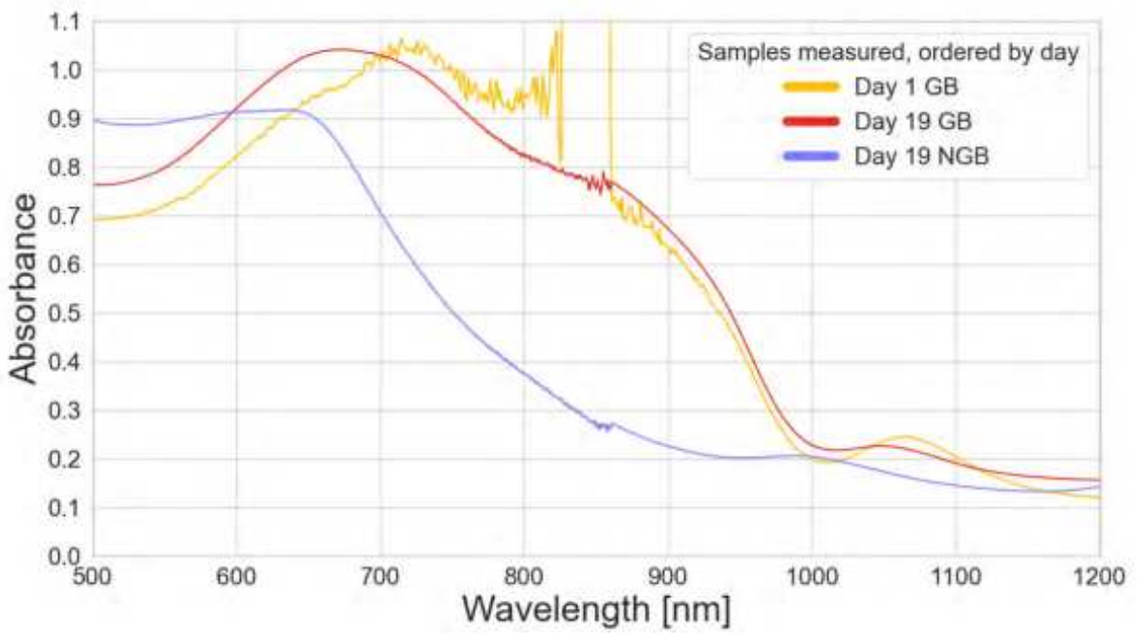


Figure 6.9: Absorbance spectrum for full-stack PSC devices with copper back electrodes. (Yellow) Day 1, fresh device. (Red) Day 19, aged device, stored inside the glovebox [GB]. (Blue) Day 19, aged device, stored outside the glovebox [NGB]. Detector defect appears around 860 nm, causing a jump in the spectra.

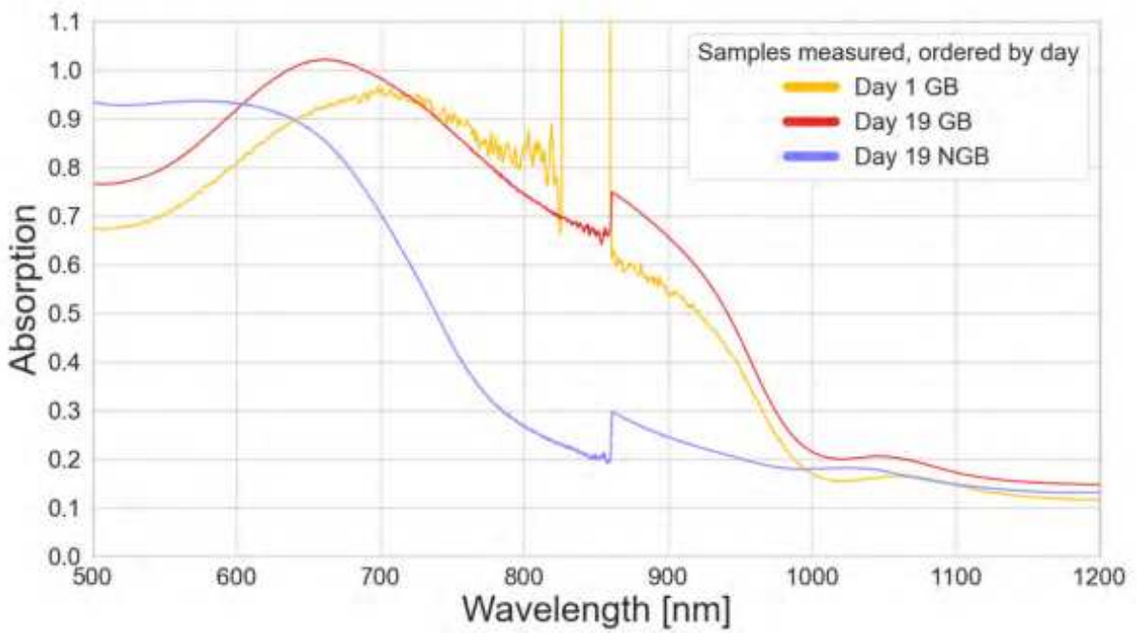
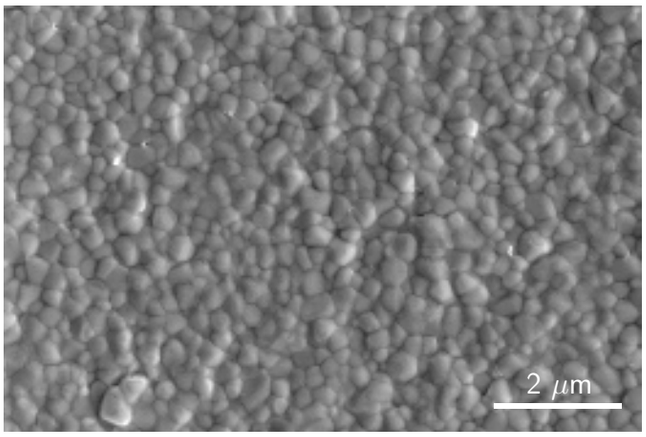


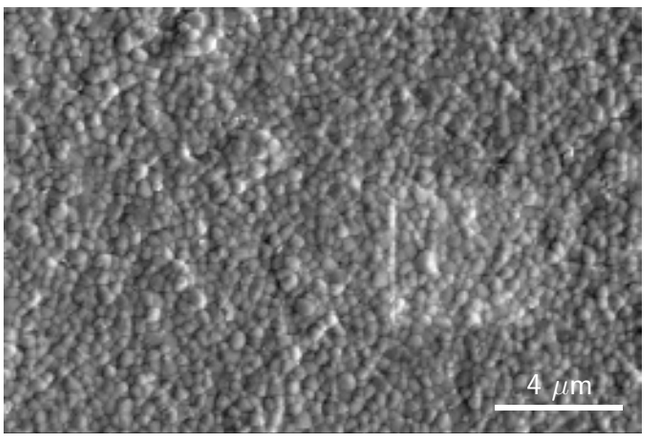
Figure 6.10: Absorbance spectrum for full-stack PSC devices with silver back electrodes. (Yellow) Day 1, fresh device. (Red) Day 19, aged device, stored inside the glovebox [GB]. (Blue) Day 19, aged device, stored outside the glovebox [NGB]. Detector defect appears around 860 nm, causing a jump in the spectra.

Then we investigated how the morphology of the samples changed. As we can see in Figure 6.11 below, the samples stored inside the glovebox went through minimal surface morphology changes. As for the samples stored in ambient air, we do see morphology changes, including the appearance of some defects

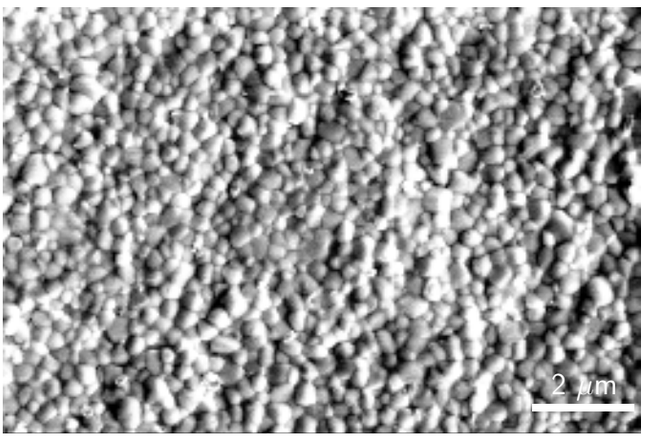
visible after only 7 days, and 28 days. These changes are in line with what was observed in the optical properties exhibited by the UV-Vis absorbance spectra, as the samples stored in the glovebox also appeared relatively unchanged.



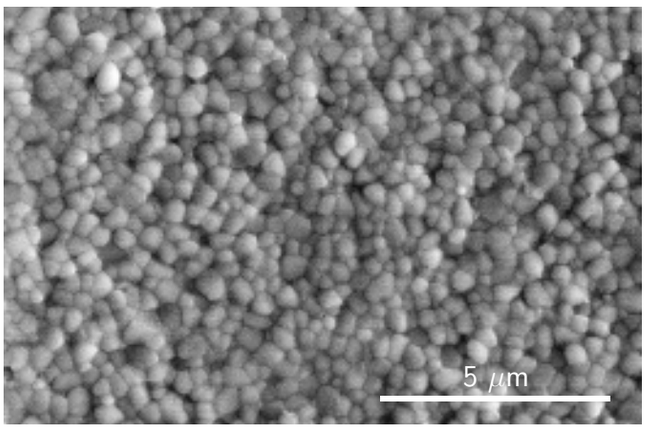
(a) Fresh sample



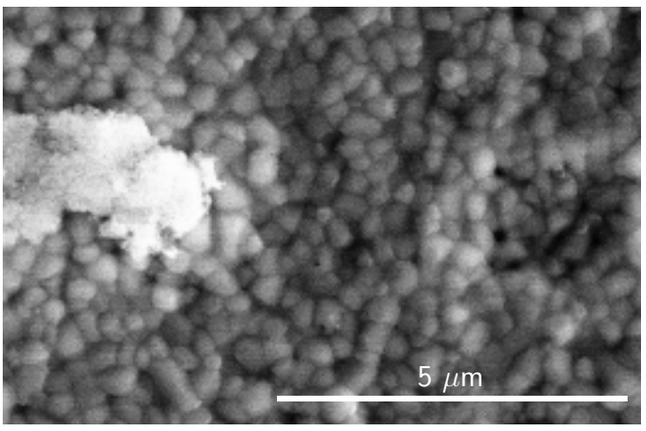
(b) Day 7 stored in glovebox



(c) Day 7 stored in ambient air



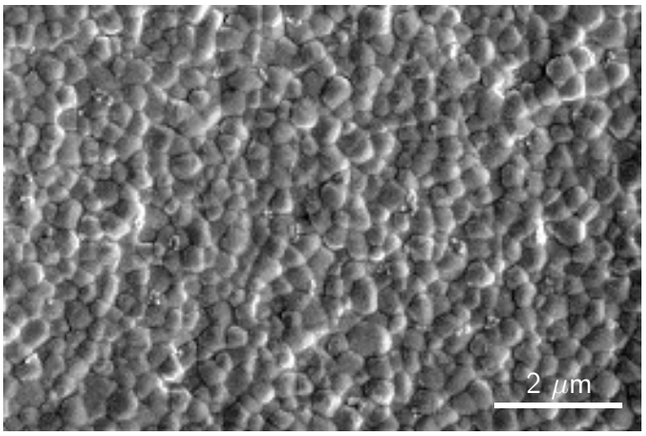
(d) Day 28 stored inside glovebox



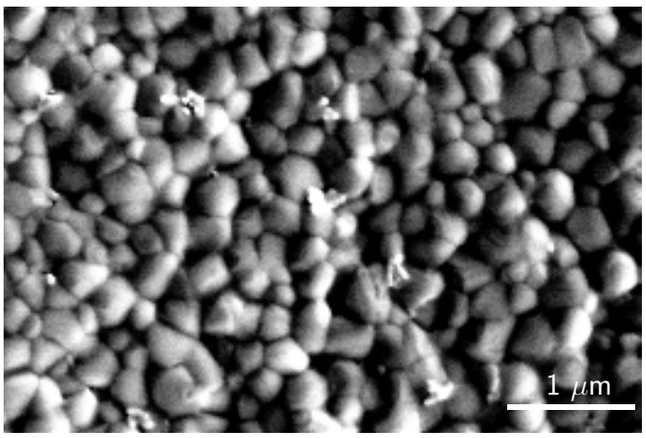
(e) Day 28 stored in ambient air

Figure 6.11: SEM images of full stack PSC devices with copper back electrodes. The samples were examined when (a) freshly made, (b)-(c) after being aged for 7 days, and (e)-(f) after being aged for 28 days.

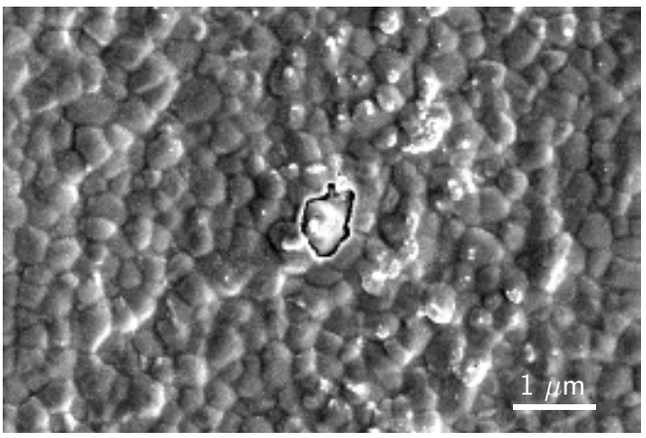
As for the silver, we see a similar effect. The samples stored inside the glovebox exhibited no new defects on their surface, and overall the morphology remained similar, albeit with a slightly rougher morphology. However, the samples stored in ambient air appear to have more of a significant change in morphology, including new defects.



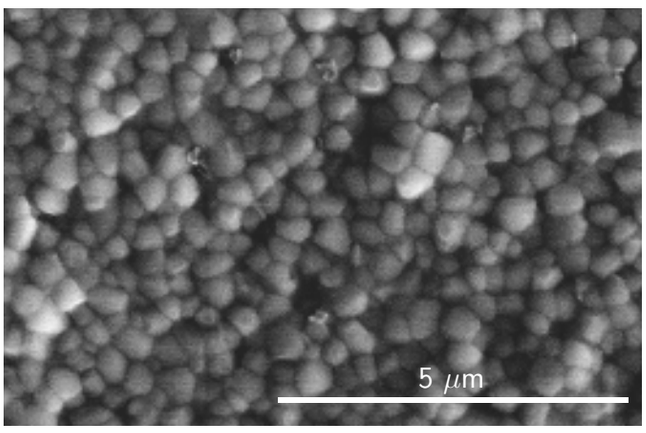
(a) Fresh sample



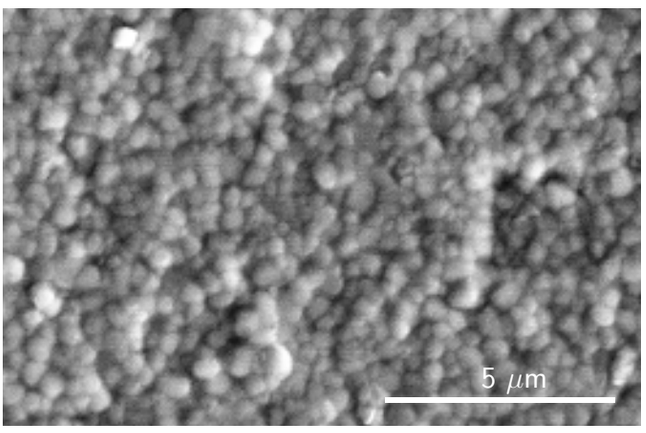
(b) Day 7, stored inside glovebox



(c) Day 7, stored in ambient air



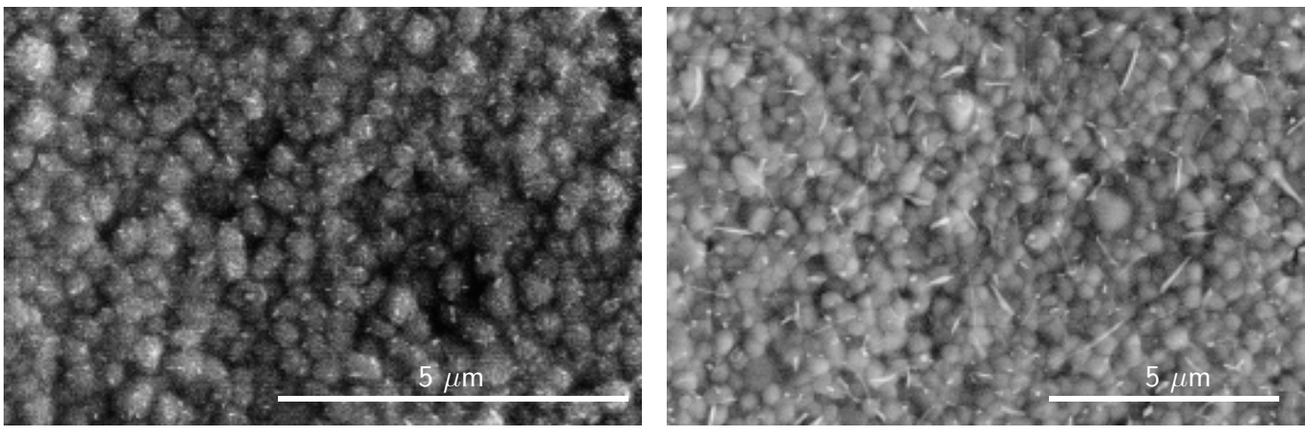
(d) Day 28, stored inside glovebox



(e) Day 28, stored in ambient air

Figure 6.12: SEM images of full stack PSC devices with silver back electrodes. The samples were examined when (a) freshly made, (b)-(c) after being aged for 7 days, and (e)-(f) after being aged for 28 days.

After 30 days, only 2 more days of ageing than the samples seen in Figures 6.11 (e) and 6.12 (e) above, the samples stored outside the glovebox showed a new type of morphological change for the full stack devices. As we can see in Figure 6.13 below, the appearance of small white rods can be seen on the surface of the samples. They appear larger in the device with silver electrodes than in the device with copper electrodes. Upon examination using EDX, we see that they cause a carbon peak in Figure 6.14 below. This is in contrast to the other elements detected in the samples, which did not have show any peaks in the areas of the rods.



(a) Day 30, stored in ambient air (Copper) (b) Day 30, stored in ambient air (Silver)

Figure 6.13: SEM images of full stack PSC devices with (a) copper and (b) silver back electrodes. The samples were examined after being aged for 30 days in ambient air.

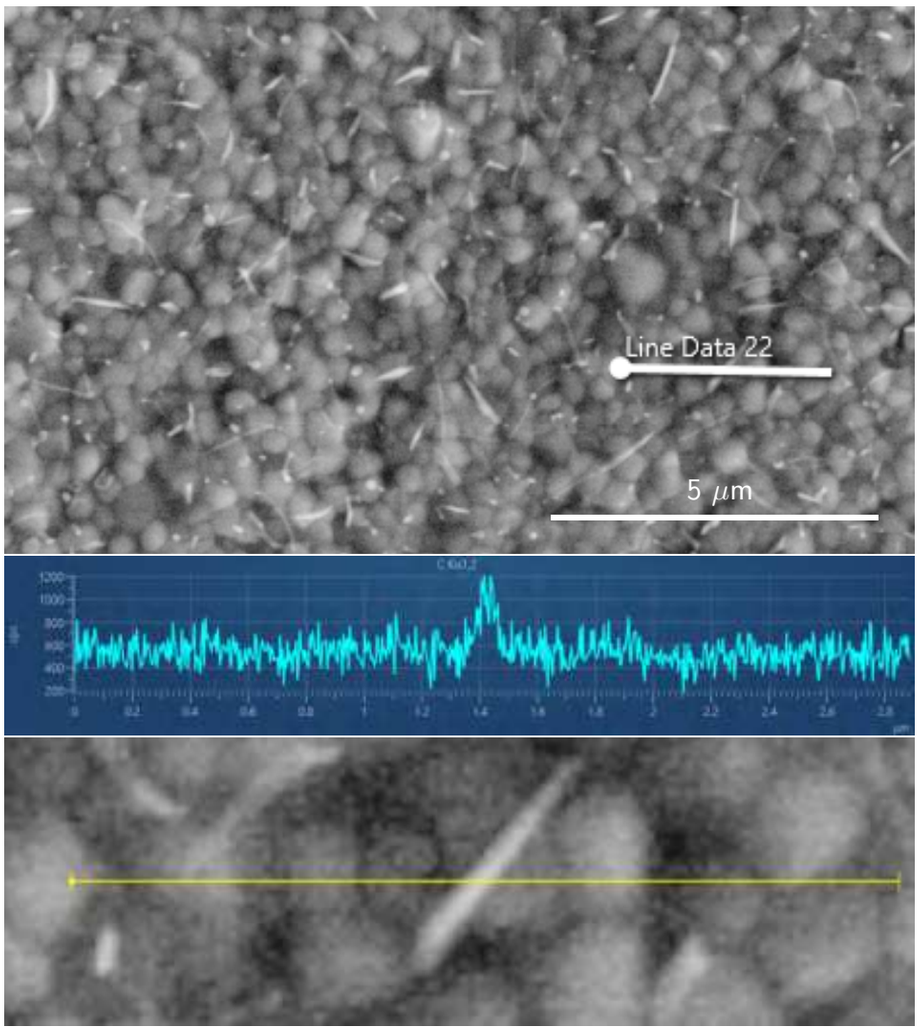


Figure 6.14: Linescan of white rod on full stack device with silver electrodes, as seen in Figure 6.13 (b) above, and the Carbon Kα1,2 peak that appears when the white rod is scanned.

In order to further understand these white rods, we examined the XRD patterns for these samples. In Figure 6.15, we see these diffraction patterns, with their perovskite peaks identified. Non-perovskite peaks

are also identified. We see that ITO peaks begin to appear for samples as they age and the absorber layer becomes thinner, both inside and outside the glovebox. In the samples stored outside the glovebox, we also see a new peak form, which can be identified as $CsSnI_3$, which is in agreement with the unpublished paper written by Rombach et al, 2024 [59]. The perovskite peaks were identified by estimation from similar perovskite structures [74, 75].

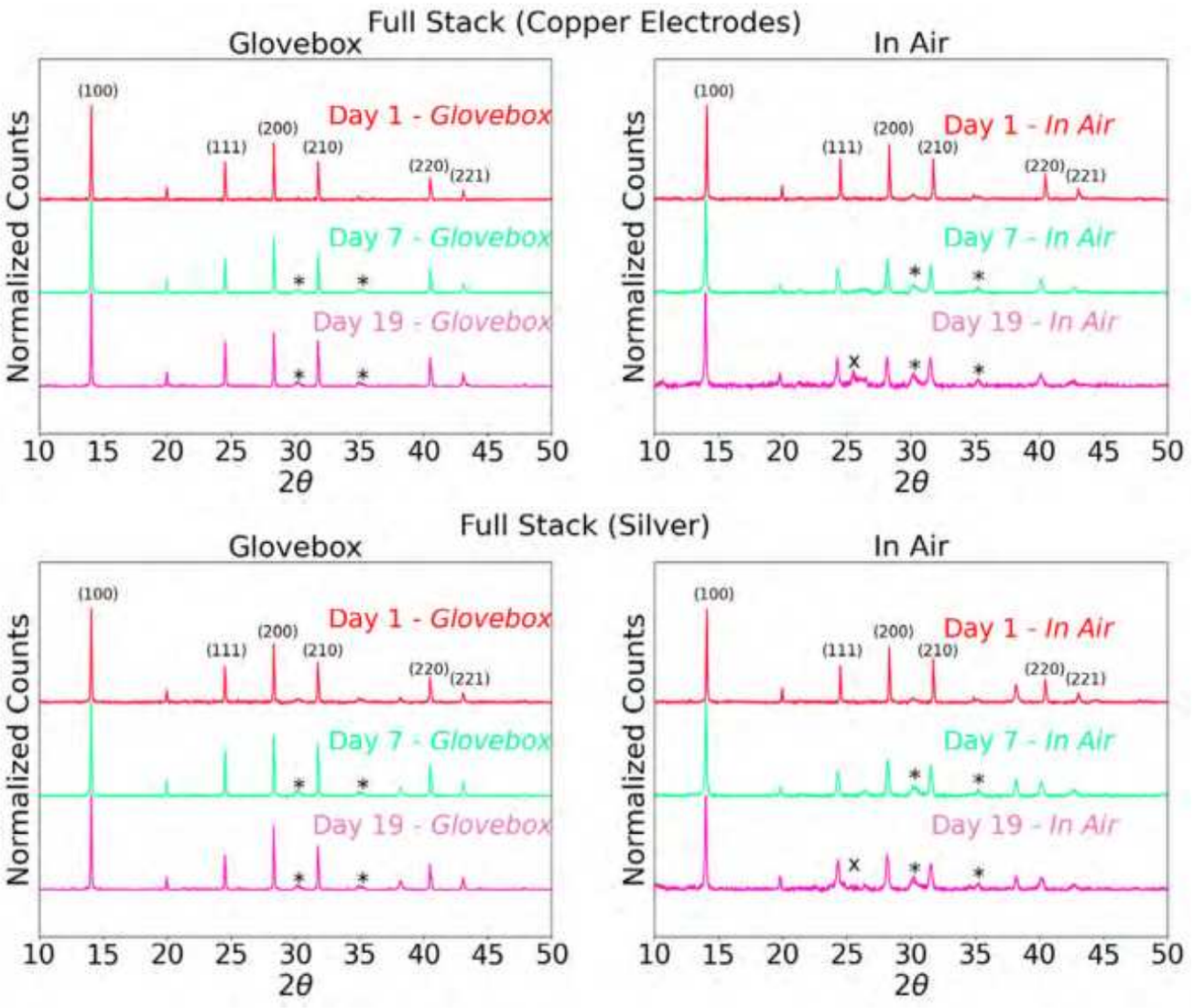


Figure 6.15: XRD patterns for full stack PSC devices with *(top row)* copper and *(bottom row)* silver electrodes, stored *(left column)* inside the glovebox and *(right column)* in ambient air. The perovskite peaks are identified, the ITO peaks are denoted by '*' and the $CsSnI_3$ peak is denoted by 'X'.

Dry air degradation

In order to further examine how these samples change, and why, the variable of moisture in the air was removed. For this part of the study, full stack devices with copper electrodes were stored inside a desiccator for 24 days and examined as described in the previous section.

We see in Figure 6.16 below how the samples' absorbance spectra changed over the course of the 24 days. These changes were similar to the samples stored in ambient air. We see a that the onset of the absorption shifts into shorter, bluer wavelengths. Similar to the devices stored in ambient air, the samples also went through a colour change, turning more red and less opaque. The yellow-tint from the electron

transport layer also disappeared over time.

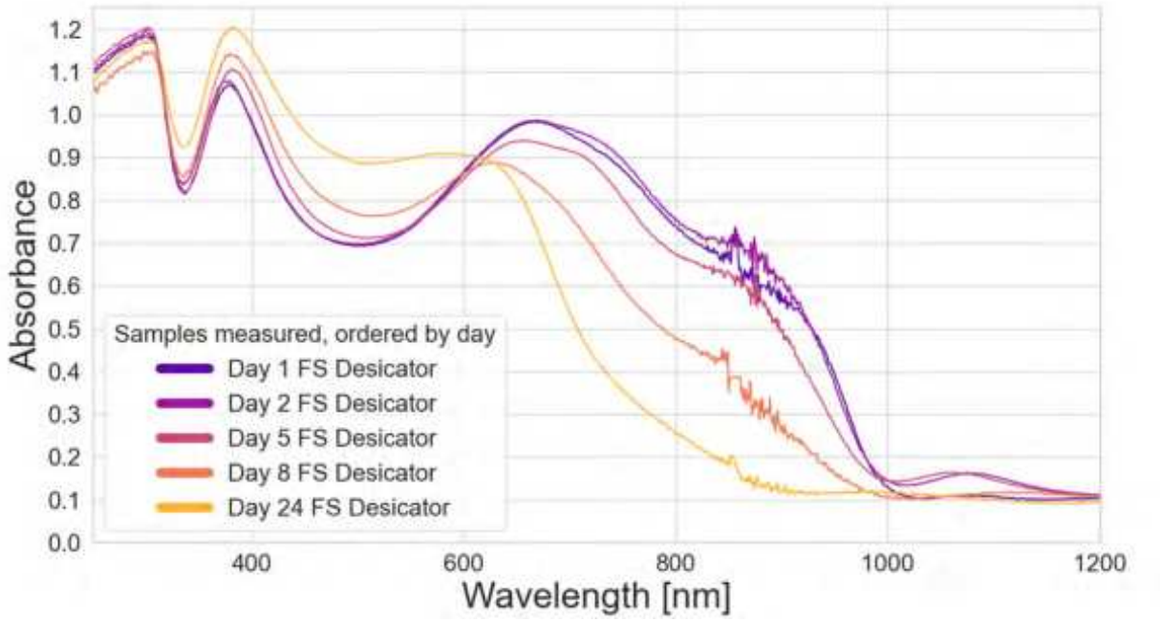
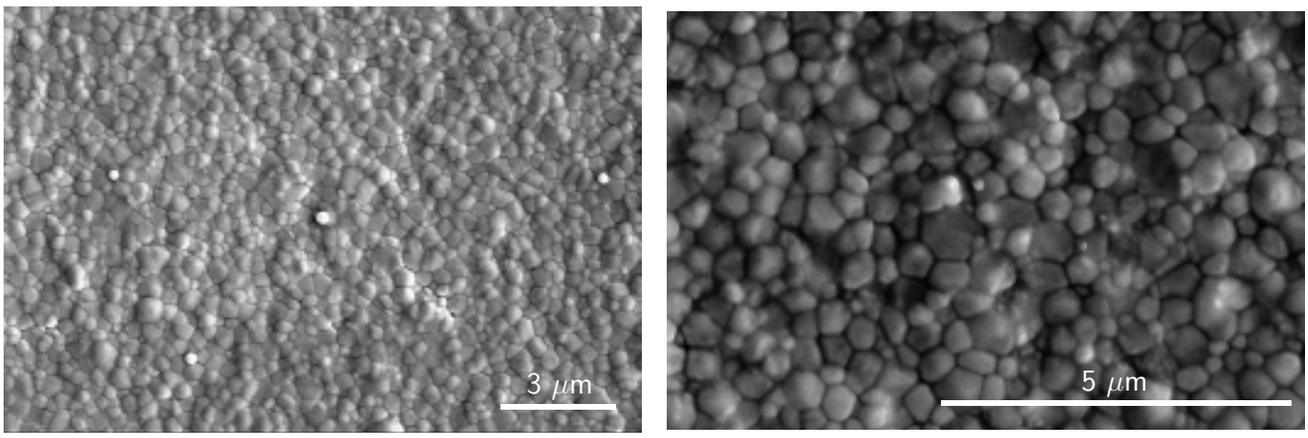


Figure 6.16: Absorbance spectra for full stack PSC devices with copper electrodes, stored in a desiccator.

The morphology of these samples was also examined, and these SEM images also showed changes similar to what was observed for the samples stored in ambient air. We see these changes for the devices stored in the desiccator for 8 and 24 days respectively in Figure 6.17 below.



(a) Full stack stored 8 days in dry air

(b) Full stack stored 24 days in dry air

Figure 6.17: SEM images of full stack PSC devices with copper back electrodes. The samples were examined after being aged for (a) 8 days and (b) 24 days in dry air.

The XRD patterns for these samples were similar to the patterns of the devices stored in ambient air. We see in Figure 6.18 below that they have nearly-identical peaks. The main difference we observe is the lack of a discernible $CsSnI_3$ peak in the samples stored inside the desiccator.

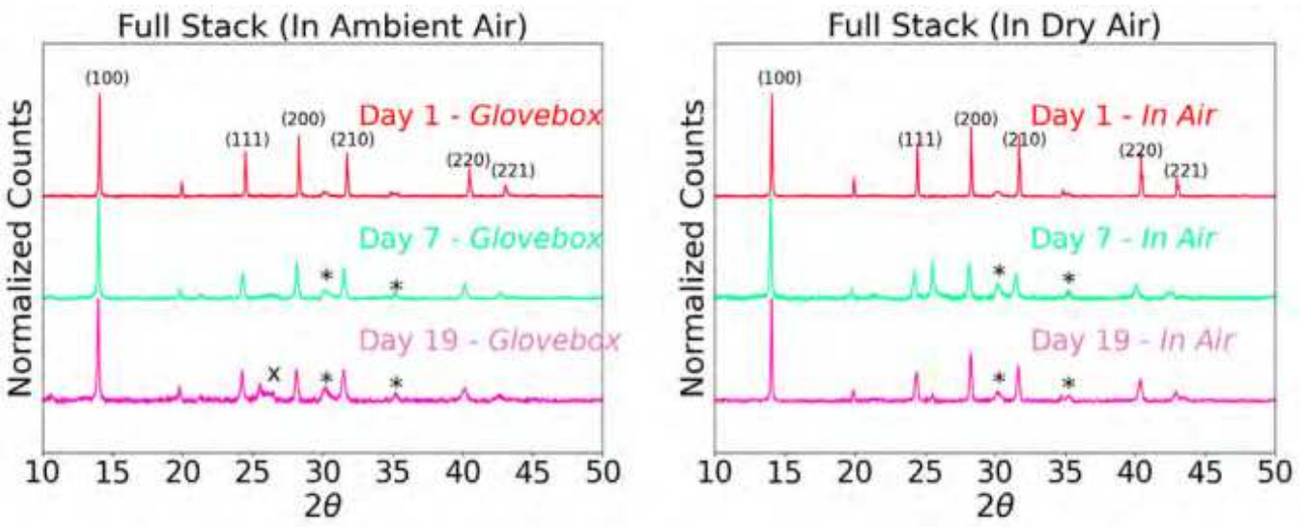


Figure 6.18: XRD patterns for full stack PSC devices with copper electrodes, stored *(left)* in ambient air and *(right)* in dry air. The perovskite peaks are identified, the ITO peaks are denoted by '*' and the $CsSnI_3$ peak is denoted by 'X'.

6.3.2 Electron Transport Layer

Next we can examine samples made without metal contacts, and without an electron transport layer to try and understand their contribution to the degradation of the solar cells. Refer to Section 4.1 for an illustration of these two partial stacks.

Ambient air degradation

The samples were stored and examined alongside the full stack devices described in the previous section. As we can see in Figures 6.19 and 6.20 below, we see the same trend as we saw with the full stack PSC devices. The samples stored inside the glovebox almost entirely retain their absorpbance spectra, whereas the samples stored in ambient air exhibit a blue-shifting of their absorption onset. The same differences were observed in these samples; that the samples stored in ambient air appear more red in colour, less opaque, and the yellow tint of the electron transport layer disappears.

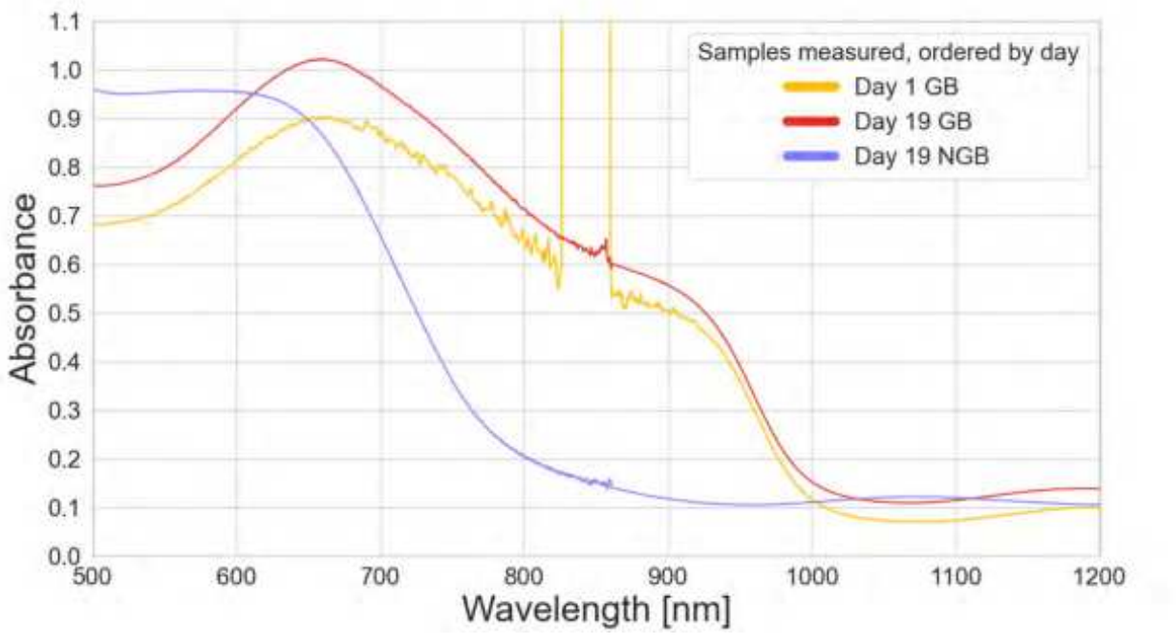


Figure 6.19: Absorbance spectrum for partial stacks without metal back electrodes. (Yellow) Day 1, fresh sample. (Red) Day 19, aged sample, stored inside the glovebox [GB]. (Blue) Day 19, aged sample, stored outside the glovebox [NGB]. Detector defect appears around 860 nm, causing a jump in the spectra.

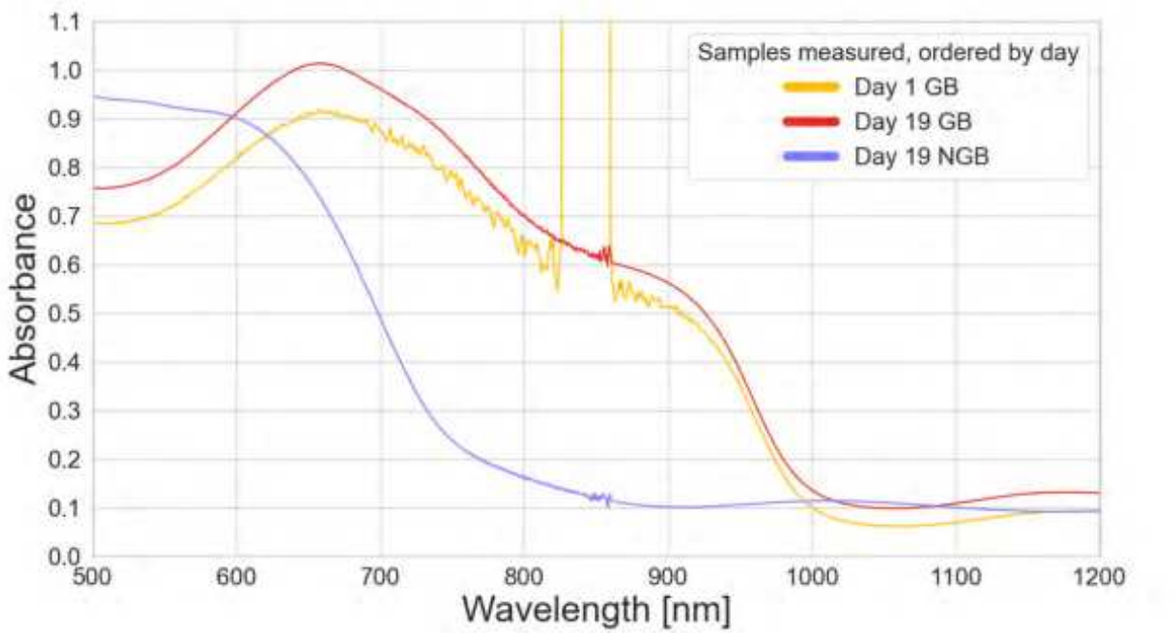
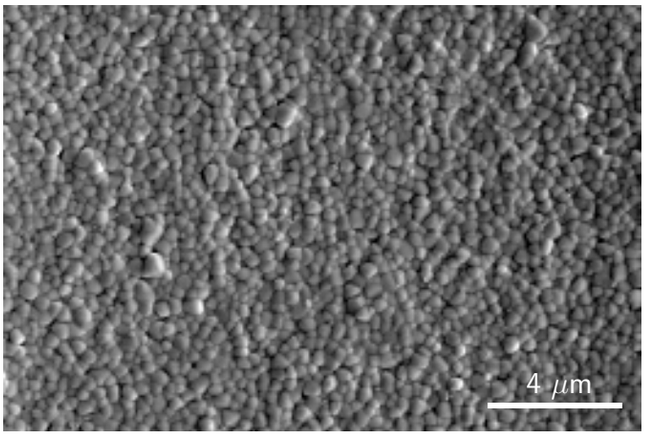


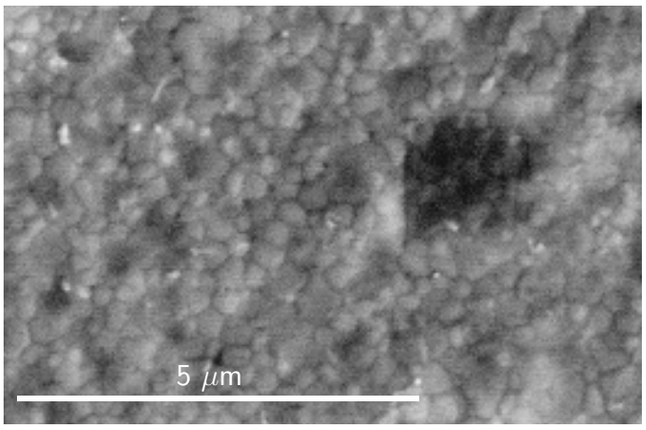
Figure 6.20: Absorbance spectrum for partial stacks without metal back electrodes or electron transport layer. (Yellow) Day 1, fresh sample. (Red) Day 19, aged sample, stored inside the glovebox [GB]. (Blue) Day 19, aged sample, stored outside the glovebox [NGB]. Detector defect appears around 860 nm, causing a jump in the spectra.

Morphological changes were examined for these samples. A clear difference is see a difference between the different stacks. We see in Figure 6.21 below, that the samples stored inside the glovebox are relatively

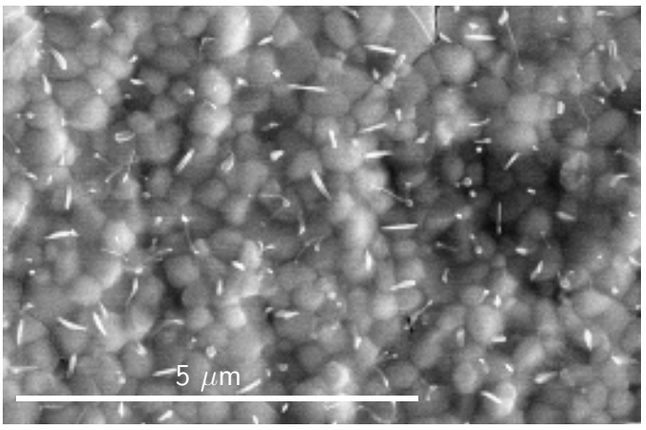
unchanged in their surface morphology, which is consistent with what we have observed in the full stack devices, and in these samples' UV-Vis absorbance spectra. However, for the samples that were stored in ambient air, we see that the white rods that appeared on the full stack devices after 30 days, are now appearing on these samples after only 7 days. Repeated experiments with these types of samples show that these rods will appear on these specific type of partial stacks in as few as 5 days stored in ambient air. These white rods were also identified as carbon using EDX, seen in Figure 6.22 below.



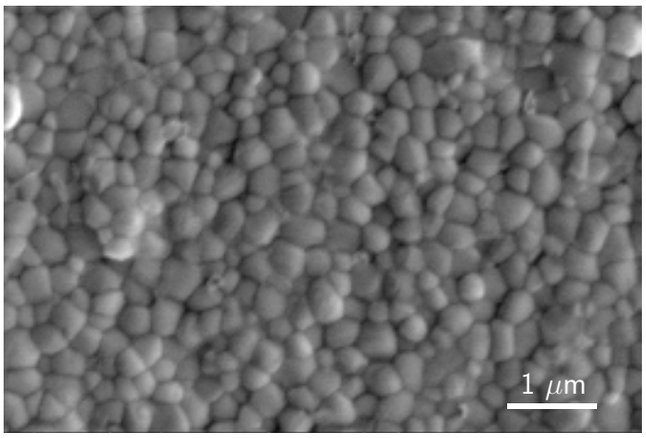
(a) Fresh sample



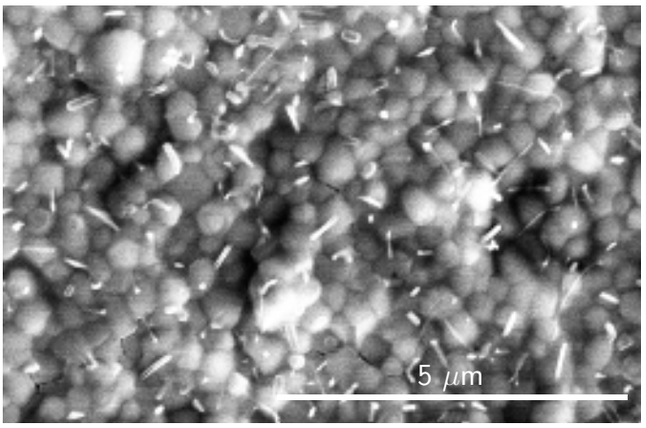
(b) Day 7, stored inside glovebox



(c) Day 7, stored in ambient air



(d) Day 28, stored inside glovebox



(e) Day 28, stored in ambient air

Figure 6.21: SEM images of partial stacks with no metal back electrodes. The samples were examined when (a) freshly made, (b)-(c) after being aged for 7 days, and (e)-(f) after being aged for 28 days.

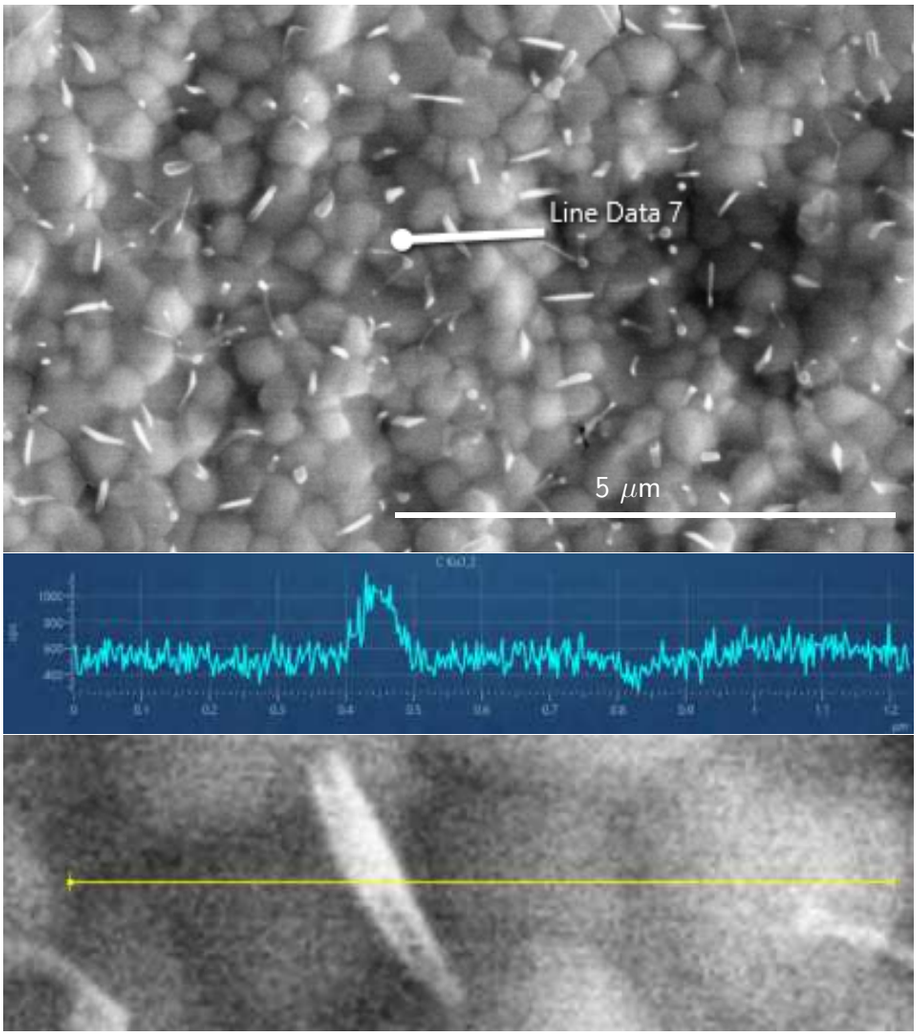
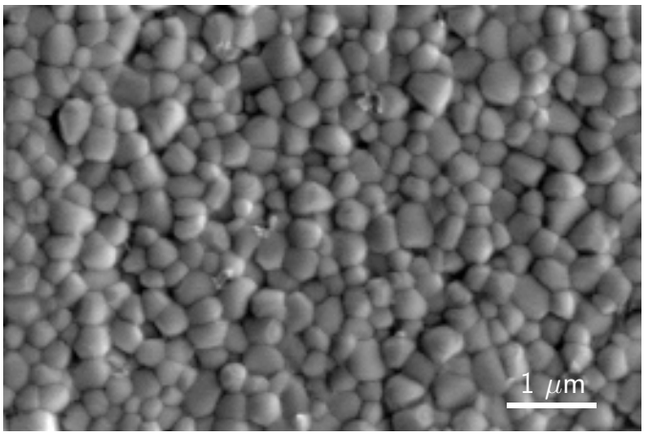
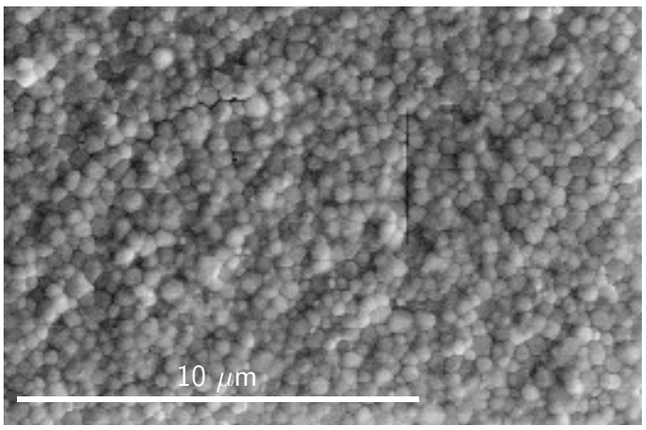


Figure 6.22: Linescan of white rod on partial stack sample with no metal electrodes, as seen in Figure 6.21 (c) above, and the Carbon Kα1,2 peak that appears when the white rod is scanned.

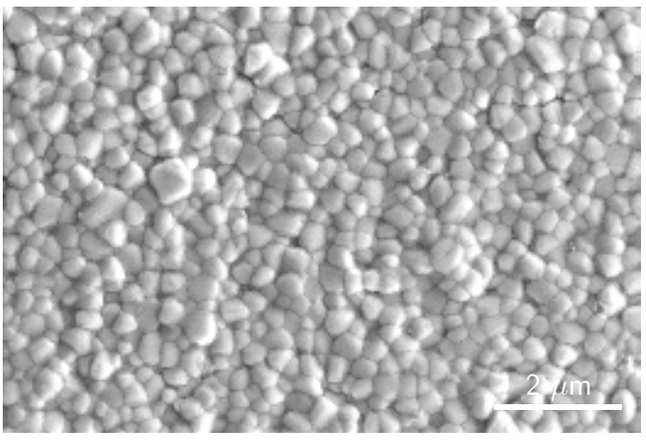
From analysing samples that have no electron transport layer, or metal electrodes, we draw the conclusion that the appearance of these white rods is shown to be related to the presence of the electron transport layer. As we can see in 6.23 below, the samples that were aged in ambient air that had no electrodes or ETL do not show the appearance of white rods, even after 28 days.



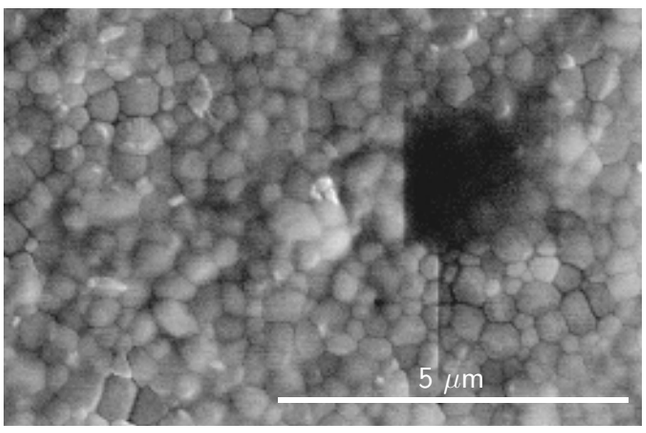
(a) Fresh sample



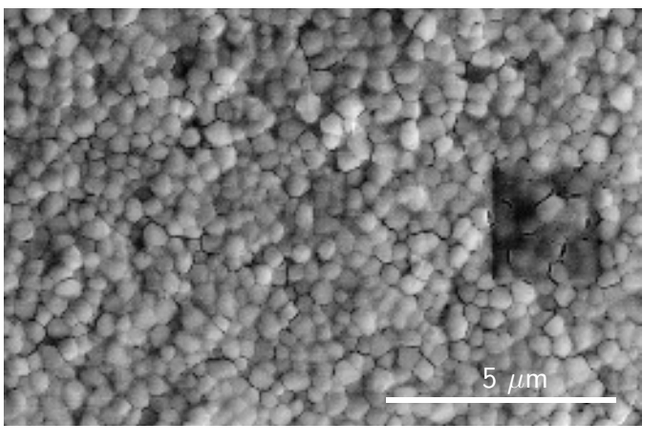
(b) Day 7, stored inside glovebox



(c) Day 7, stored in ambient air



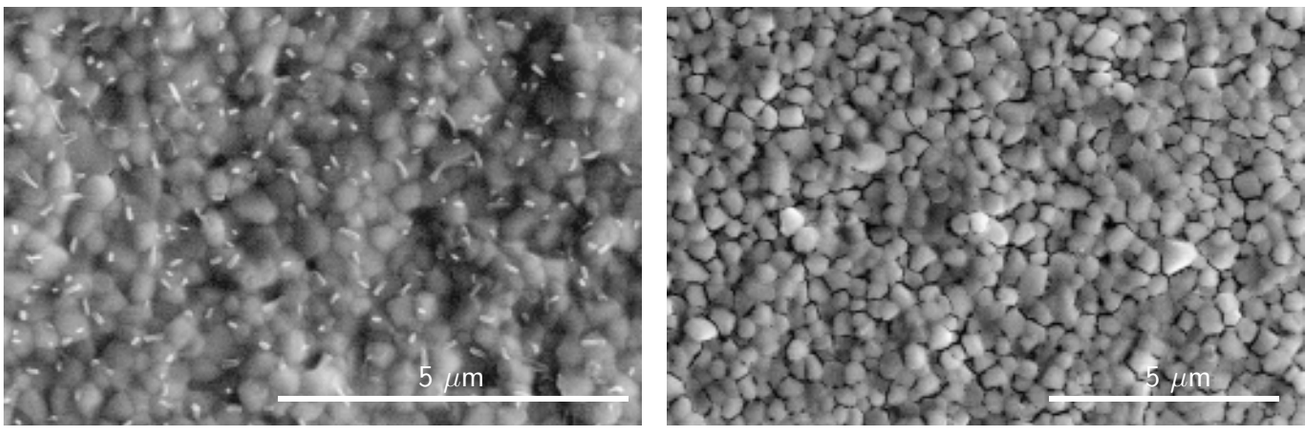
(d) Day 28, stored inside glovebox



(e) Day 28, stored in ambient air

Figure 6.23: SEM images of partial stacks with no metal back electrodes and no electron transport layer. The samples were examined when (a) freshly made, (b)-(c) after being aged for 7 days, and (e)-(f) after being aged for 28 days.

To further solidify this theory, the samples were also imaged in the SEM after being aged in ambient air for 30 days, as this was when the white rods appeared for the full stack devices (refer back to Figure 6.13). In Figure 6.24 below, we see that the sample with an ETL does show these white rods on its surface, whereas the sample without the ETL does not, even after 30 days.



(a) Day 30, stored in ambient air (b) Day 30, stored in ambient air

Figure 6.24: SEM images of partial stack PSC devices with (a) no metal electrodes and (b) no ETL and no metal electrodes. The samples were examined after being aged for 30 days in ambient air

We examined the XRD patterns for these samples over the intermittent time intervals over the degradation period. As we can see in Figure 6.25 below, there are no unique peaks on the XRD patterns that could indicate a new crystalline structure that would represent the white rods.

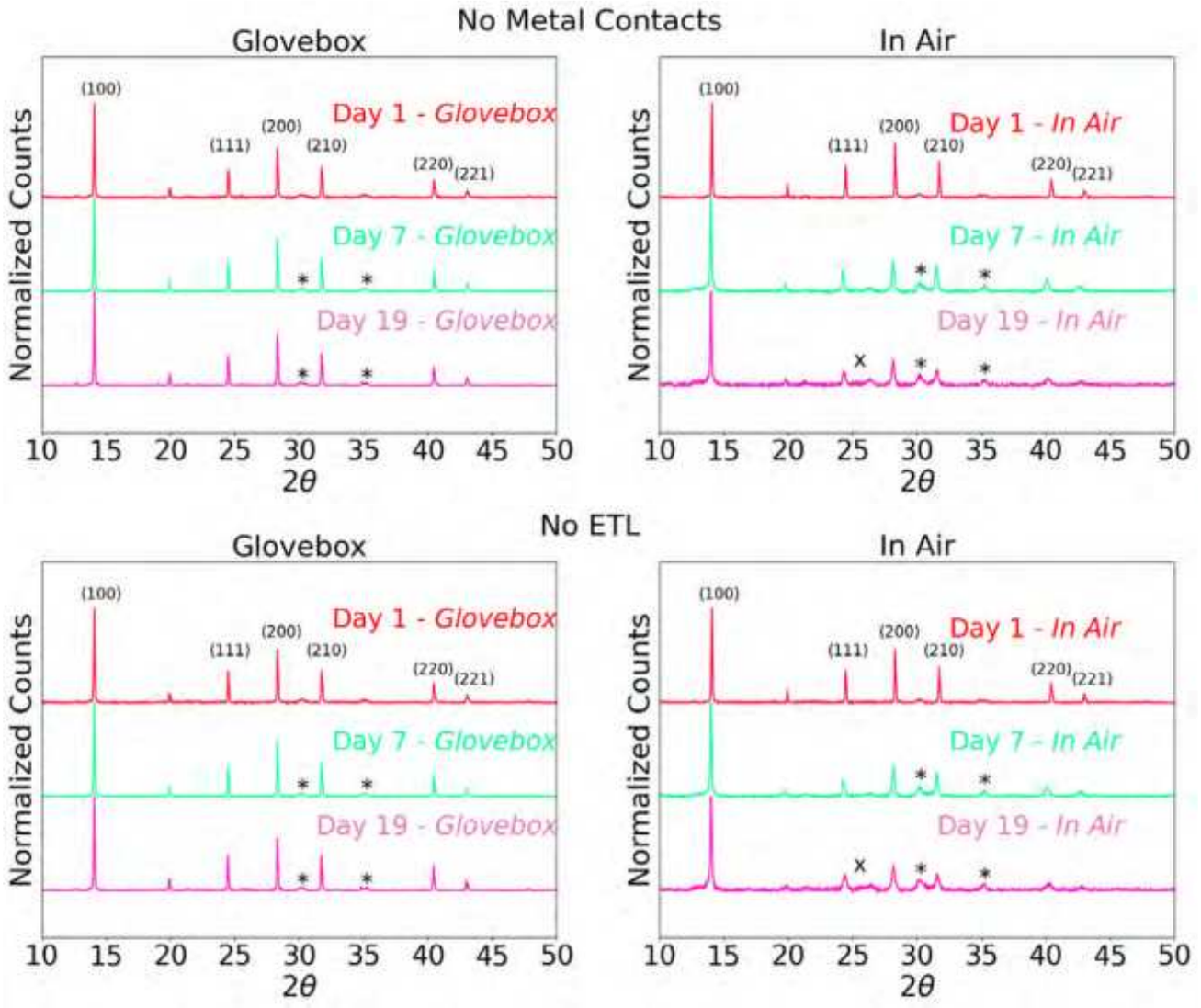


Figure 6.25: XRD patterns for partial stacks with *(top row)* no metal electrodes and *(bottom row)* no ETL or metal electrodes, stored *(left column)* inside the glovebox and *(right column)* in ambient air. The perovskite peaks are identified, the ITO peaks are denoted by '*' and the $CsSnI_3$ peak is denoted by 'X'.

Dry air degradation

The same characterisation measurements were performed on the same type of samples, with half of them stored in dry air in a desiccator for 24 days. This batch included the full stack devices discussed previously. This was to glean more information on the origins of the degradation process that produces the amorphous carbon white rods.

As we can see in Figures 6.26 and 6.27 below, they follow the same trends we saw before in the samples stored in ambient air with the changing of the absorption onset, as well as a similar trend to the full stack devices stored inside the desiccator. The visual change in the samples was also consistent, where they turned less opaque and more reddish in colour, with the yellow tint from the samples with the ETL disappearing over time.

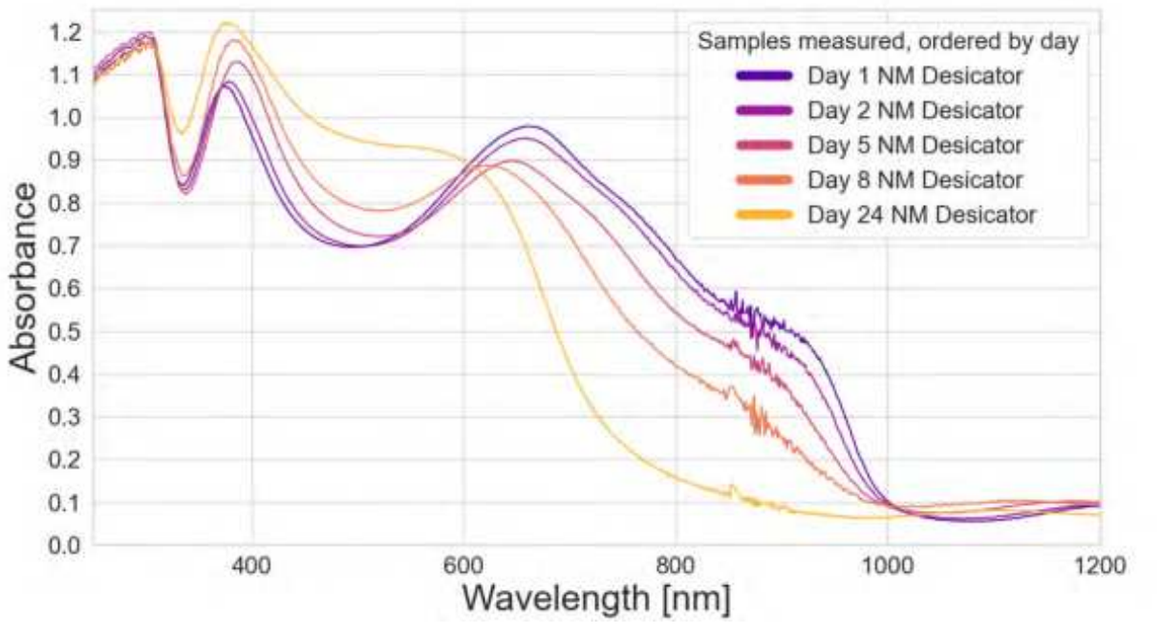


Figure 6.26: Absorbance spectrum for partial stacks with no metal back electrodes, stored in a desiccator, over the course of 24 days.

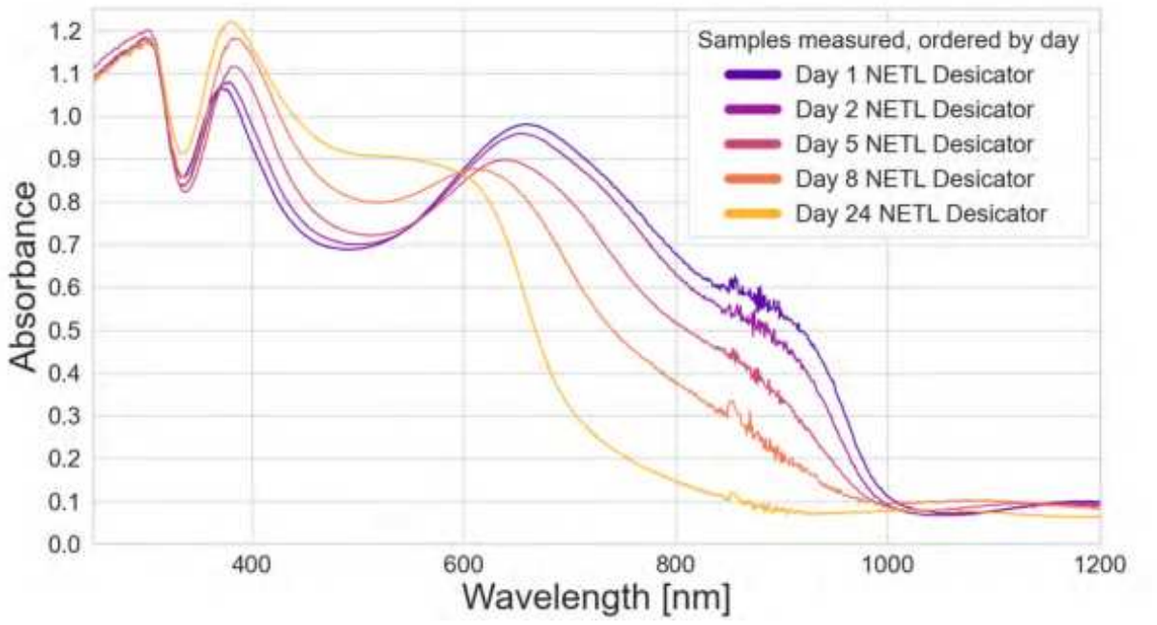
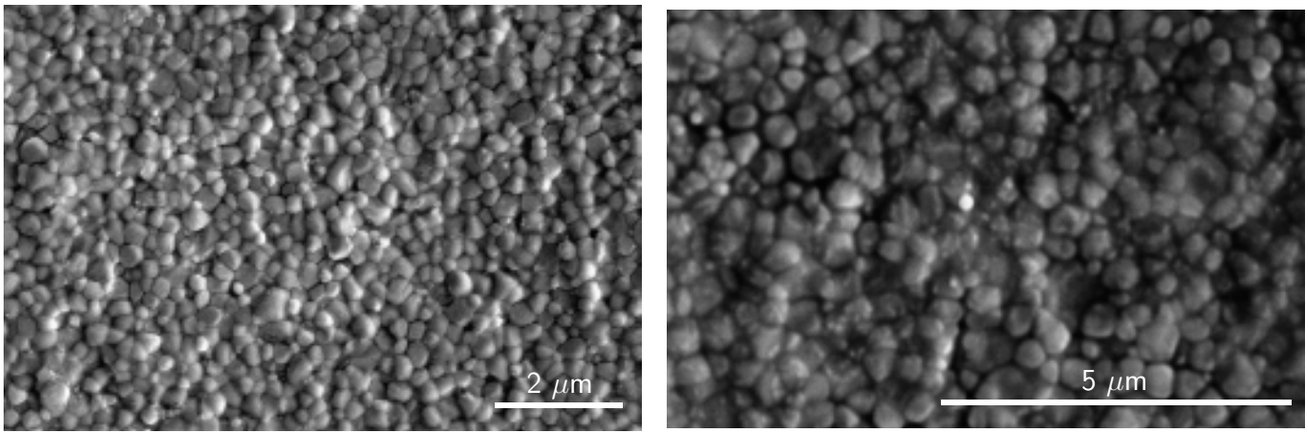


Figure 6.27: Absorbance spectrum for partial stacks with no ETL and no metal back electrodes, stored in a desiccator, over the course of 24 days.

Once again, the surface morphology of these samples was examined over the course of their degradation period. Below in Figure 6.28 we see a different result than expected. Where the samples with an ETL and no electrodes had shown the appearance of white rods on their surface when stored in ambient air for a minimum of 5 days (replicated over the course of a few different batches), these samples had no appearance of white rods, even after 24 days. This further develops the theory on how the white rods

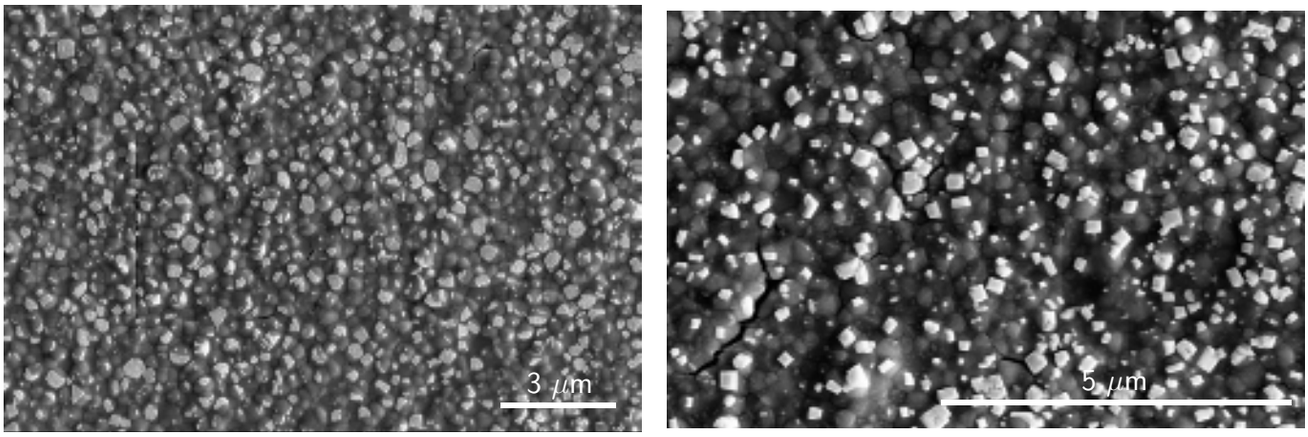
appear on the surface, as it does not seem to be a purely oxygen-driven process, and instead is a process that requires the samples to be exposed to moisture.



(a) Day 8, stored in dry air (b) Day 24, stored in dry air

Figure 6.28: SEM images of partial stack PSC devices without metal back electrodes. The samples were examined after being aged for (a) 8 days and (b) 24 days in dry air.

Additionally, the samples without an ETL also showed a different surface morphology than what was observed when stored in ambient air. We can see this in Figure 6.29 below. These samples appeared to have a bright cubic-like structure form on its surface. When these cube-like regions were examined with an EDX, we saw no unique element peaks, leading to a theory that it may be a secondary cubic perovskite phase forming on its surface.



(a) Day 8, stored in dry air (b) Day 28, stored in dry air

Figure 6.29: SEM images of full stack PSC devices without an ETL or metal back electrodes. The samples were examined after being aged for (a) 8 days and (b) 24 days in dry air.

We examined the XRD patterns of these samples to compare them to their counterparts stored in ambient air, which can be seen in Figure 6.30 below. We did not observe any unique peaks appearing in the samples stored inside the desiccator. The main difference between the two batches is, as observed in the full stack devices, the presence of the $CsSnI_3$ peak in the samples stored in ambient air, whereas there was no such peak in the samples stored in dry air.

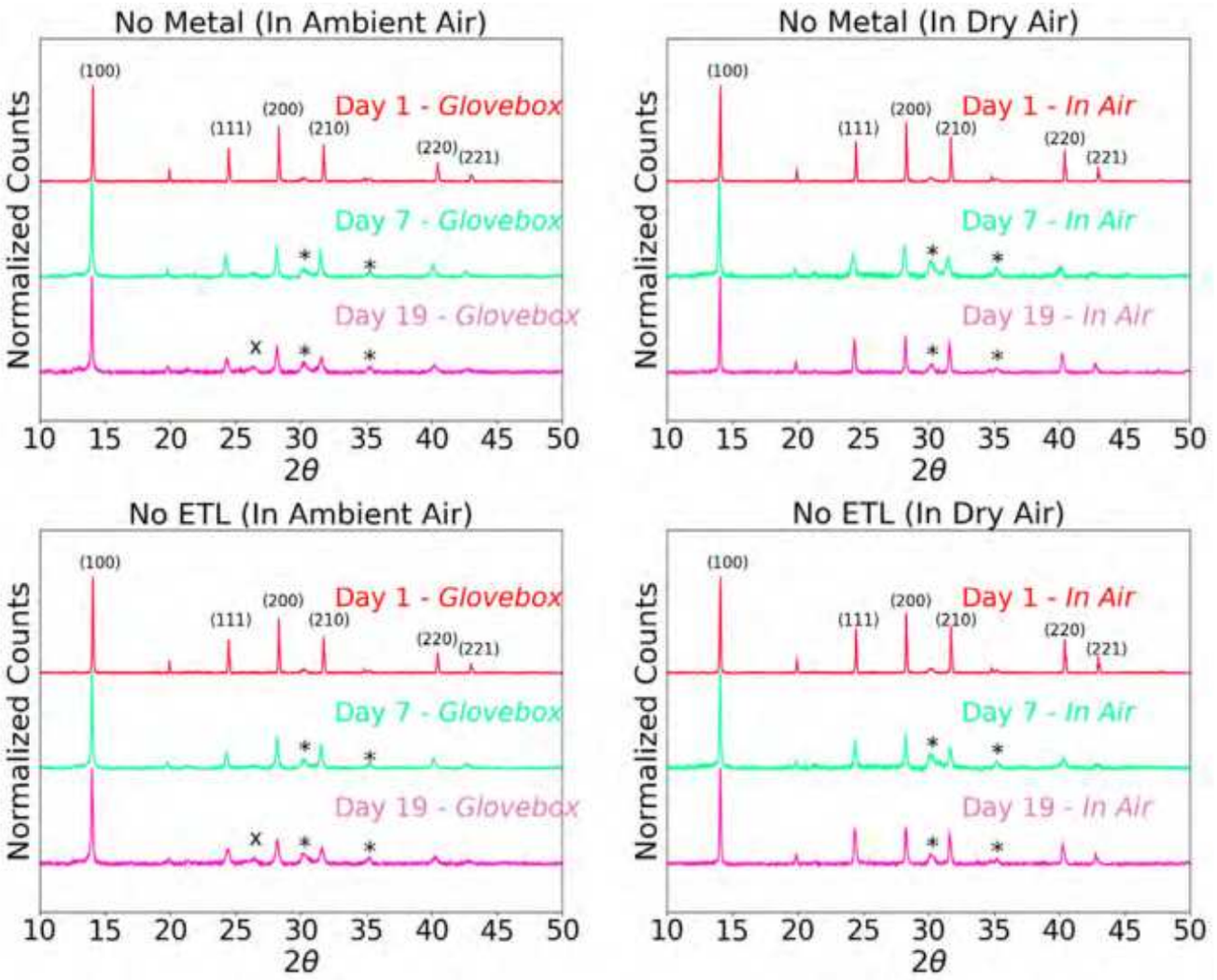


Figure 6.30: XRD patterns for partial stacks with *(top row)* no metal electrodes and *(bottom row)* no ETL or metal electrodes, stored *(left column)* in ambient air and *(right column)* in dry air. The perovskite peaks are identified, the ITO peaks are denoted by '*' and the $CsSnI_3$ peak is denoted by 'X'.

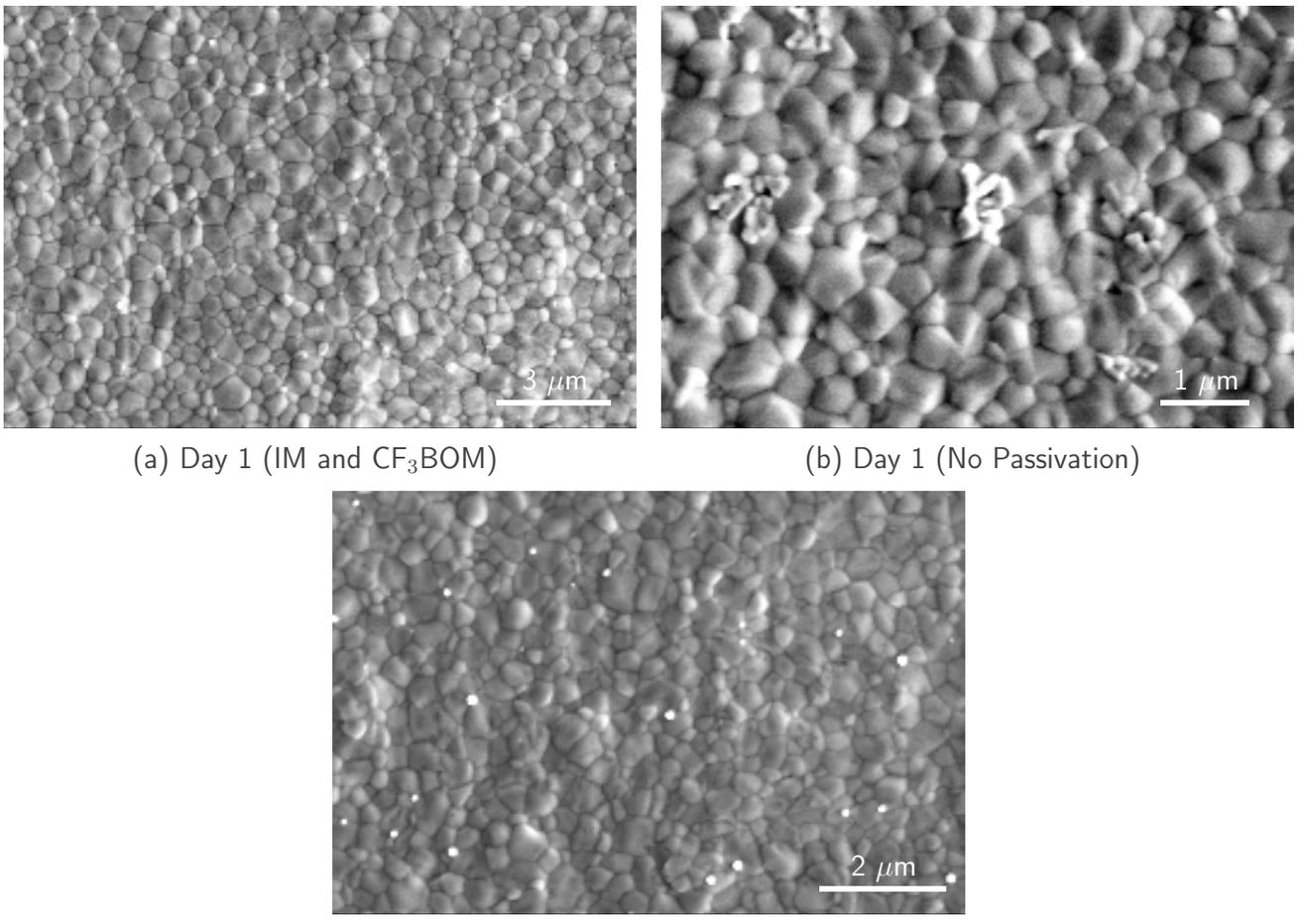
6.3.3 Passivation Layer

It can be clearly seen that the perovskite/ETL interface creates a unique form of degradation in comparison to the other interfaces. In order to further examine this, we altered this interface. As was mentioned previously, a passivation layer was deposited between the perovskite and electron transport layer, in order to improve their contact and reduce interfacial defects. For all previous samples discussed so far, and an EDAI monolayer was used for the passivation layer.

Short-term degradation experiments were repeated for samples with a different passivation layer (IM and CF_3BOM) (see Chapter 4 for details), and without a passivation layer. These samples were aged in two ways by storing them in ambient air as we had done for the previous batches.

Ambient Air Degradation

In ambient air, we see very similar results as we had seen for the full stack samples with the EDAI passivation. Below in Figure 6.31 (c) we can see that the full stack devices with the IM/CF_3BOM passivation layer also do not show white rods after 7 days of ageing. Below in 6.31 (a), we do see small white spheres on the surface, but these were present in the fresh samples. EDX analysis identifies these as tin-based spheres. We also see in the devices with no passivation layer, in Figure 6.31 (b), flower-like structures. These were identified with EDX analysis as SnF_2 (see Appendix E), which is in line with the flower-like structures reported in Liao et al's 2016 paper [76]. No white rods appeared after 7 days on these samples either, however due to data loss from the server no image is available to exhibit this.



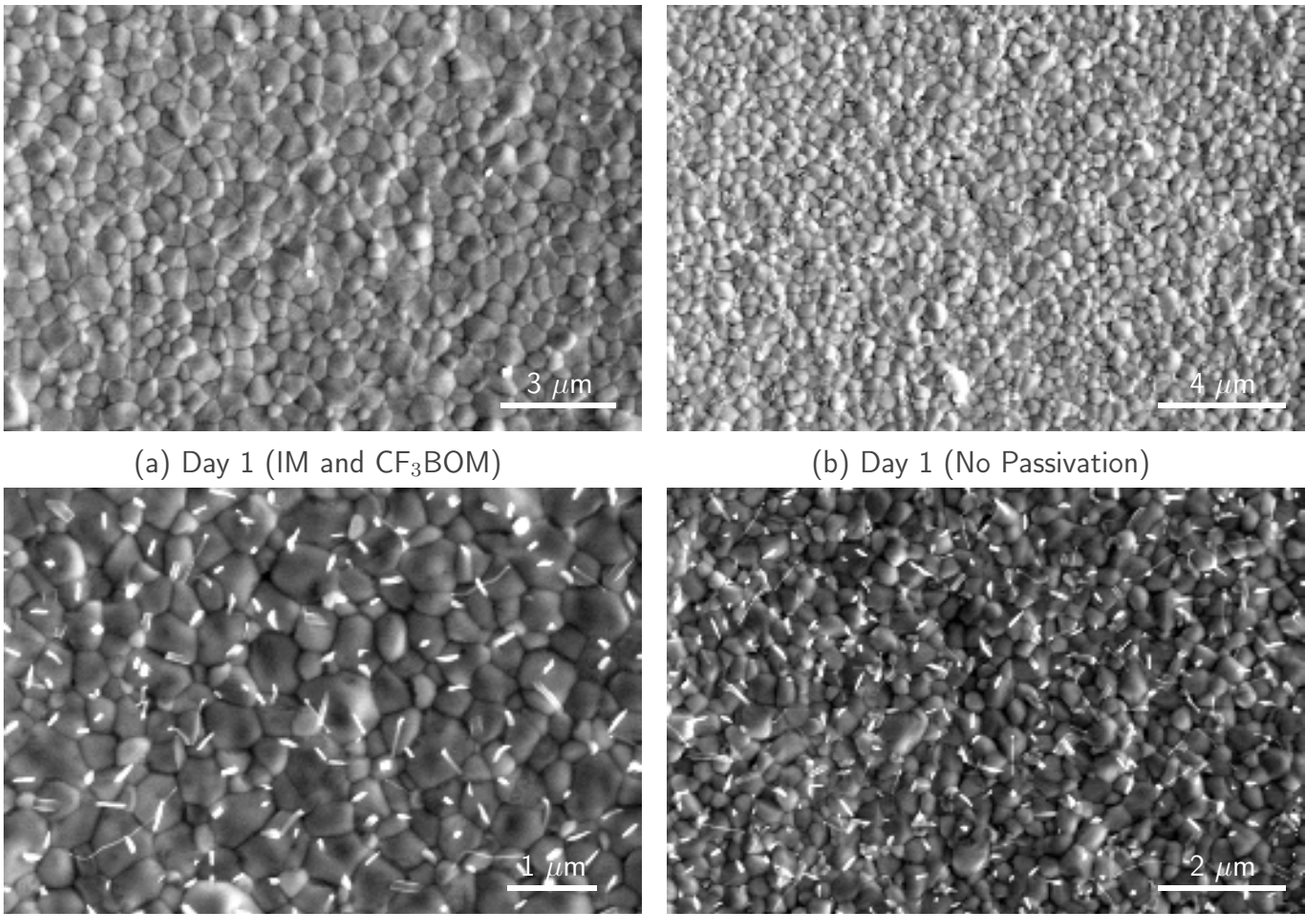
(a) Day 1 (IM and CF_3BOM)

(b) Day 1 (No Passivation)

(c) Day 7 (IM and CF_3BOM)

Figure 6.31: SEM images of full stack devices with (a & c) a IM/CF_3BOM passivation layer and (b) no passivation layer. The samples were examined when (a-b) freshly made, (c) after being aged for 7 days.

In the samples without metal contacts we also see the same results in the IM/CF_3BOM passivation layer samples and the no passivation layer samples, as we saw in the samples with the EDAI passivation layer. Below in Figure 6.32 we see that both the alternate passivation layer and the samples without passivation exhibit the white rods. These rods were once again identified as carbon using EDX analysis.



(a) Day 1 (IM and CF_3BOM)

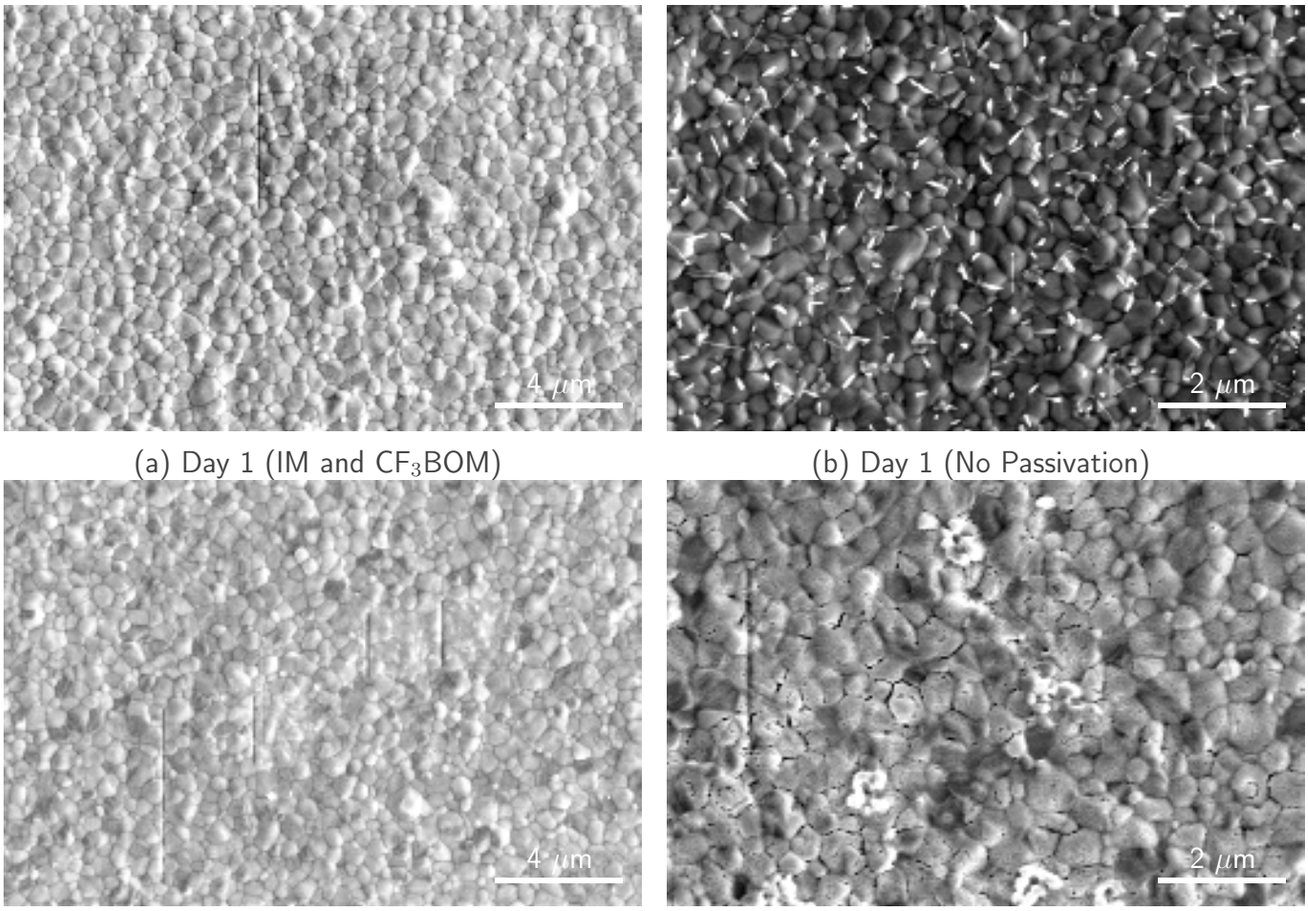
(b) Day 1 (No Passivation)

(c) Day 7 (IM and CF_3BOM)

(d) Day 7 (No Passivation)

Figure 6.32: SEM images of partial stacks with with (a & c) a IM/CF_3BOM passivation layer and (b & d) no passivation layer, as well as no metal back electrodes. The samples were examined when (a-b) freshly made, (c-d) after being aged for 7 days.

Finally, we see that the samples without metal contacts and without an ETL exhibit the same results as before with a changed passivation layer. As we can see in Figure 6.33 below, these samples did not show any evidence of white rods on their surface. Both of these types of samples appeared very beam-sensitive, particularly the aged sample without a passivating layer. Focusing the beam was found to cause cracks appearing at the grain boundaries. For the samples without passivation, the flower-like structures were present for these partial stacks as well.



(a) Day 1 (IM and CF_3BOM)

(b) Day 1 (No Passivation)

(c) Day 7 (IM and CF_3BOM)

(d) Day 7 (No Passivation)

Figure 6.33: SEM images of partial stacks with with (a & c) a IM/CF_3BOM passivation layer and (b & d) no passivation layer, as well as no metal back electrodes or electron transport layer. The samples were examined when (a-b) freshly made, (c-d) after being aged for 7 days.

6.3.4 Discussion

When we examine the UV-Vis absorbance spectra of all the different stacks, we see that the optical properties of the samples seem to degrade in a very similar way. The samples stored inside the glovebox appear to retain their same absorbance spectra as when they are freshly made, indicating that the optical properties do not significantly change in the absence of moisture and oxygen. Thus, the changing of this property appears to be mostly dependant on their exposure to oxygen, as the trend is consistent in ambient and dry air (see section 4.8). The samples aged in ambient and dry air showed a change in their spectral shape along the same trend: a blue-shift of the absorption onset. None of the samples appear to exhibit a different trend of degradation unique to their stack category, and appear to degrade along roughly the same timescale. The samples without ETL did appear to visually become more translucent faster, and more red. An explanation for this could be that the surface of the perovskite is more exposed to the atmosphere, due to there being no extra layer over the perovskite to act as a barrier for oxygen and moisture. To further investigate this in future studies, more frequent absorbance measurements could be taken on the samples, in shorter time frames, e.g. measuring the absorbance over 15 minute - 1 hour intervals, to see if the initial response to exposure to ambient air is different. Concurrent photoluminescence measurements may also shine more light on how the optical properties of each type of sample changes, and be a good compliment to the absorbance measurements.

Similarly we see in our XRD pattern analysis that the perovskite peaks for all samples stored in an oxygenated environment experience a peak shift to the left (see Appendix H) . This is in line with what has been observed for CsFA lead-tin perovskites in literature [75]. The shifting of the peaks could be due to a small increase in the perovskite crystals' lattice parameter [58] or due to a change in lattice strain [77]. Further investigation into the origins of this peak shifting could be done in order to expand the understanding of the potential degradation mechanisms at play here. We can also see that over time the ITO peaks increase in relative intensity, which gives credence to the assumption that the perovskite layers might be thinning. This is not seen in the samples stored in the inert, nitrogen environment of the glovebox. Their peaks appear to be at the same 2θ position, and we do not see increased intensity in the ITO peaks.

The composition of the white rods that appear on the surface of aged samples was investigated. In a study by Liao et al in 2016, there were similar defects seen on the surface of pure-tin perovskite ($FASnI_3$) cells, identifying them as related to the concentration of SnF_2 in the perovskite precursor [76]. Similarly, a study done by Pious et al in 2023 also observed bright rod-like defects on the surface of lead-tin perovskite ($(FASnI_3)_{0.6}(MAPbI_3)_{0.4}$) cells. These were also related to the SnF_2 concentration in the perovskite precursor solution. They found that at concentrations of SnF_2 higher than 5 mol %, the SnF_2 will segragate at both the HTL/perovskite and ETL/perovskite interfaces. This can then result in easily oxidised SnF_2, forming $SnO_{1.2}F_{0.2-0.5}$ at the surface. [78] In both of these studies, they identified tin and fluorine at these small, bright, rod-like structures using EDX. When we performed EDX analysis on the samples, we ran map scans and line scans across multiple regions of these white rods, over the course of multiple batches. Repeatedly we did not detect any tin peaks or fluorine peaks, but we did consistently detect strong carbon emission peaks from these regions. This was detected using a variety of accelerating voltages (2.5 keV - 12.5 keV). The maps and linescans were typically run between 3 - 15 minutes, depending on the sample. Perovskite can be very beam-sensitive, particularly the degraded samples. Therefore, we had to be careful to balance getting the highest resolution possible for the measurements with not damaging the samples too egregiously. In Appendix F we can see the aftermath of running these measurements on the surface of a perovskite sample.

These results were different to what we found in the studies that identified their white rods as SnF_2. Firstly, these bright rod structures were observed for them in freshly made samples, not aged ones. Secondly, while the structures look similar to the bright rods we observe on our samples, our EDX analysis provided us with different results. We ensured that SnF_2 can be correctly identified, as these defects were identified

correctly in the samples seen in Figure 6.31 (see Appendix E). Our EDX analysis identified these rods as carbon. Furthermore, a lack of unique peaks in the XRD pattern analysis, across the different types of samples, is also noted. This was unexpected as white rods appeared on the surface of the samples with an electron transport layer over time. Therefore, the interpretation of the XRD patterns is that the carbon rods are amorphous.

We now need to examine the perovskite stack, and try to determine where this amorphous carbon was coming from. we managed to rule out external contaminants, as all 3 types of the samples were stored in the same sample boxes and in the same environments. we knew that two likely sources of carbon were either within the perovskite layer, or from the electron transport layer. Within the perovskite layer, there was formamidinium iodide ($HC(NH_2)_2I$). This was an obvious source of carbon. As for the electron transport layer, which is C_{60} and BCP ($C_{26}H_{20}N_2$), both materials contain carbon. From literature review and considering the changes in each of the samples, we formed the theory that the amorphous carbon was coming from the C_{60} section of the electron transport layer. This was due to a few reasons. Firstly, these white rods only appear on samples that have an electron transport layer. The samples that have no ETL still have the perovskite and passivation layers, which would contain formamidinium. The samples that had no passivation layer or an alternative passivation layer (with an ETL) also exhibited these morphology changes, indicating that the carbon does not come from the passivation layer materials. Secondly, through literature review, we was unable to find evidence that formamidinium typically degrades into any form of amorphous carbon in ambient conditions, with a study involving $FAPbI_3$ suggesting that the distrubution of FA ions does not change to exposure of light or humidity alone, but instead needs a combined stress [79]. Particularly CsFA compositions for tin-lead perovskites have been found to be significantly stable under damp-heat conditions [57]. Thirdly, it is known that fullerenes have poor morphological stability due to fullerene diffusion and aggregation in the thin film over time, caused by their relatively weak intermolecular interactions [63, 80]. Thirdly, while BCP layer is unstable, both in air and under thermal stress. BCP exposed to air quickly aggregates and crystallises [66]. This means it does not create amorphous carbon, but it does expose the layers below it to infiltration of moisture and oxygen.

The timescale before the appearance of the amorphous carbon rods is significantly elongated by the presence of an ETL/metal electrode interface, wherein the timeline of the morphology change lengthens from $\sim$5 days without the presence of the metal electrode interface to $\sim$30 days with the presence of the metal electrode interface. An interfacial interaction that could be considered as an explanation for the appearance of the amorphous carbon rods is the potential mixing of the C_{60} and BCP, which allows the metal to come into direct contact with the C_{60} [45], which could change the mechanisms occuring at the perovskite interface. It has been seen that this metal diffusion can be accelerated in operating PSCs, which could be triggered by the J-V curve measurements. We could alternatively explain this to be the copper acting as a desiccant for the perovskite surface, which has been reported for fully encapsulated PSCs by Wijesekara et al in 2021 [68]. The delay could indicate that the copper metal contacts are being preferentially corroded by the oxygen and moisture in the ambient atmosphere, demonstrating that moisture may play a role in this degradation process.

The humidity of the environment in which the samples were stored and aged was observed to make a substantial difference to how the surface morphology changed. The samples that were stored in ambient air experienced $\sim$72% humidity and the samples stored in the desiccator experienced $\sim$16% humidity. We checked this using a hydrometer, which was stored in the same environment as the samples for about 12 hours each. The desiccator was not completely sealed, and so the moisture was not completely removed from the air, but still provided a substantial difference in humidity for the purposes of this experiment. The samples with and ETL and no metal contacts that were stored in dry air did not show any white rods on the surface, even after 24 days, despite their ambient air counterparts doing so after 5-7 days. These samples without an ETL and metal contacts stored in dry air also showed a new bright cube-like surface. This was assumed to be a perovskite phase, as we did not observe any particular EDX element peaks or unique XRD peaks for this surface. Therefore, the morphological change we observed appeared

to be related more to the presence of moisture and oxygen, than just oxygen alone.

In deeper investigation into how this may relate to a potential degradation mechanism of C_{60}, we discovered a study that had been done on organic photovoltaics (OPV) solar cells, where C_{60} had been investigated as an ETL in a planar *p-i-n* structure stored in both ambient and dry air. This study, done by Bastos et al in 2016, concluded that the degradation of C_{60} is primarily driven by oxygen, although the presence of moisture accelerates this [81]. Further, there have been studies done on the possibility of amorphisation of C_{60} to give amorphous carbon products, albeit more focused on high-temperature thermal degradation and not perovskite-focused [82]. This still gives some credence to the theory that under certain stresses, such as exposure to moisture and oxygen, enough stress may cause the breakdown of the fullerene cage and formation of amorphous carbon by-products.

Chapter 7: Outlook and Conclusion

Through the use of the SCIPRIOS SpinBot we have successfully improved the reproducibility of perovskite solar cells within a batch, and between batches. We can see that the average performance of a device has become more consistent within a batch. This is useful for qualitative studies on perovskite thin film samples and solar cell devices, allowing for more conclusive comparisons. Further study on how the different fabrication variables (tip height, annealing temperature/time, dispense speed of solvent) affect the overall performance and consistency of batches would help develop the understanding and improve the performance even further. In particular, improving the atmosphere of the spincoating environment further, for example through introducing more nitrogen flushing into the spincoating bowl, may significantly improve the perovskite optoelectronic properties. These potentially higher quality films/devices then may be more stable in both the long- and short-term.

The improvement in reproducibility allowed us to draw good comparisons between different perovskite stacks, by keeping the fabrication of the perovskite absorber layer and passivation layer as consistent as possible. We used this to compare how different methods of device ageing give rise to different types of degradation of the perovskite stacks. This included light-soaking to examine the photodegradation of the PSC devices, and ageing in ambient and dry air atmospheres.

The light-soaking and biasing measurements showed us that the operational stability of a lead-tin PSC stack under illumination and applied biases can be influenced by the interface of the perovskite absorber layer and the electron transport layer. By changing the passivation layer at this interface, we see different behaviours in the devices. The devices were also shown to be stable during the measurement period when no stress was applied, showing that PSC devices can be degraded by exposure to light and under electrical biasing. Since only one round of these measurements was done, further repetitions of these measurements should be done to check the reproducibility of these results. Repetitions of these measurements and combining them with other investigations, such as concurrent impedance experiments, will help elucidate the degradation pathways. Longer light-soaking cycles may also expand the understanding on how stable the cells are under light exposure, and their ability for recovery. Imaging of the surface morphology of these samples that were aged under light may be an interesting way to expand on this study in the future.

In the investigation into the contribution of the electron transport layer and metal contact interfaces to the degradation of the overall lead-tin perovskite solar cell, we saw that the perovskite/ETL interface causes a change in the morphology of the sample. In samples that had this interface, SEM imaging showed that amorphous carbon rods appear when aged in ambient air. The timescale of the appearance of these rods was influenced by the ETL/metal contact interface. The presence of the rods after ageing was also seen to be influenced by the humidity of the environment they were stored in, indicating that the process is driven, or accelerated by, moisture. In order to investigate this degradation mechanism further, and understand the origins of this degradation pathway, repetitions of the dry air ageing experiments should be done, for a longer timescale. Measurements on these white rods could be done using XPS to increase the understanding of their composition. Cross-section measurements of the samples with white rods could also be done to investigate if these carbon rods only appear on the surface, or if there is any migration into the perovskite layer. To study the influence of these rods on the operational stability of the solar cells, fresh electrodes could be evaporated onto samples where they are present, and J-V measurements taken.

These ageing experiments may also be expanded to alternative, non-fullerene electron transport layer materials, to substitute C_{60}, to act as a control to cement the theory that these rods are originating from

the ETL and not the perovskite material itself. Stacks may also be made with a direct perovskite/metal contact interface and aged.

Overall, there are still many pathways through which the contribution of interfaces to perovskite solar cell degradation could be investigated. Through this, we may glean a better understanding of the stability of PSCs and propel this into a commercially viable solar PV technology.

Acknowledgements

I would like to thank my supervisor and group leader, Bruno, for your sage advice, patience, and encouragement. Thank you for taking any shame away from having "stupid questions", and for fostering such a warm and welcoming culture in the group. I thoroughly enjoyed getting to be a part of your team, and getting the best possible experience of being a real scientist.

I would also like to thank my daily supervisor, Daphne, for being incredibly supportive, kind, and for helping me learn so much more than I ever thought I could. I loved sharing an office with you, and Chris, and I will miss our daily chats immensely. Getting to work with you on this project has made me feel so much more capable as a scientist, thanks to your unwavering support. I am so happy to have started this project with a mentor, and have ended it with a friend.

Also, a huge thank you the rest of the Hybrid Solar Cells team (past and present): Imme, Larissa, Moritz, Jeroen, Lars, Agustin, Marc, Jarla, Tae-Youl, Sebastian, Dimitris, and Paul. Each one of you have helped me on countless occasions, from letting me practise presentations, helping with lab equipment mishaps, and just being absolutely delightful people who I feel lucky to have gotten to spend (nearly) every day with for the past 10 months.

I am grateful for my friends and family. My parents, Carolanne and Michael, who have done nothing but be in my corner completely, and without whom I couldn't have possibly gotten this far in life, let alone finish this Master's degree. Thank you for all the phone calls and cheerleading. My fellow master's students who I have been so lucky to have met: Mareike, Marina, Johanna, Ana, Jon, George, Fernando, Martín, Jonne, and Theo. Thank you for being such fantastic people.

I am also thankful to my partner, Önder, who has been my rock through many exhausting periods of late nights in the lab and impending deadlines. I will forever be filled with gratitude for your warmth and love during this important period of my life.

Bibliography

1. Lindsey, R. *Climate Change: Atmospheric Carbon Dioxide | NOAA Climate.gov* Apr. 2024. https://www.climate.gov/news-features/understanding-climate/climate-change-atmospheric-carbon-dioxide.
2. NOAA. *Global Monitoring Laboratory - Carbon Cycle Greenhouse Gases* 2024. https://gml.noaa.gov/ccgg/trends/gl_trend.html.
3. United Nations Climate Change. *New Analysis of National Climate Plans: Insufficient Progress Made, COP28 Must Set Stage for Immediate Action | UNFCCC* Nov. 2023. https://unfccc.int/news/new-analysis-of-national-climate-plans-insufficient-progress-made-cop28-must-set-stage-for-immediate.
4. IPCC. *The evidence is clear: the time for action is now. We can halve emissions by 2030. — IPCC* Apr. 2022. https://www.ipcc.ch/2022/04/04/ipcc-ar6-wgiii-pressrelease/.
5. IEA. *Tripling renewable power capacity by 2030 is vital to keep the 1.5°C goal within reach – Analysis - IEA* July 2023. https://www.iea.org/commentaries/tripling-renewable-power-capacity-by-2030-is-vital-to-keep-the-150c-goal-within-reach.
6. U.S. Department of Energy - Office of Energy Efficiency and Renewable Energy. *Solar Photovoltaic Technology Basics* https://www.energy.gov/eere/solar/solar-photovoltaic-technology-basics.
7. Rajagopal, A., Yao, K. & Jen, A. K. Toward Perovskite Solar Cell Commercialization: A Perspective and Research Roadmap Based on Interfacial Engineering. *Advanced Materials* **30.** ISSN: 15214095 (Aug. 2018).
8. IEA. *Solar - IEA* July 2023. https://www.iea.org/energy-system/renewables/solar-pv.
9. IEA. *Renewable electricity capacity additions by technology and segment, 2016-2028 – Charts – Data & Statistics - IEA* Jan. 2024. https://www.iea.org/data-and-statistics/charts/renewable-electricity-capacity-additions-by-technology-and-segment-2016-2028.
10. Laalioui, S. *Perovskite-based solar cells : from fundamentals to tandem devices* ISBN: 9783110760606 (2022).
11. Zhang, W. *et al.* Revealing key factors of efficient narrow-bandgap mixed lead-tin perovskite solar cells via numerical simulations and experiments. *Nano Energy* **96.** ISSN: 22112855 (June 2022).
12. Li, B. & Zhang, W. Improving the stability of inverted perovskite solar cells towards commercialization. *Communications Materials 2022 3:1* **3,** 1–13. ISSN: 2662-4443. https://www.nature.com/articles/s43246-022-00291-x (Sept. 2022).
13. Yadegarifard, A. *et al.* FA/Cs-based mixed Pb–Sn perovskite solar cells: A review of recent advances in stability and efficiency. *Nano Energy* **112.** ISSN: 22112855 (July 2023).
14. Queisser, H. J. Detailed balance limit for solar cell efficiency. *Materials Science and Engineering: B* **159-160,** 322–328. ISSN: 0921-5107 (Mar. 2009).
15. LMPV. *Detailed Balance (DB) Charts – LMPV* 2023. https://www.lmpv.nl/db/.
16. Savill, K. J., Ulatowski, A. M. & Herz, L. M. *Optoelectronic Properties of Tin-Lead Halide Perovskites* July 2021.

17. Al-Ashouri, A. *et al.* Monolithic perovskite/silicon tandem solar cell with >29% efficiency by enhanced hole extraction. *Science* **370,** 1300–1309. ISSN: 10959203. https://www.science.org/doi/10.1126/science.abd4016 (Dec. 2020).

18. Hu, S. *et al.* Narrow Bandgap Metal Halide Perovskites for All-Perovskite Tandem Photovoltaics. *Chemical Reviews*. ISSN: 0009-2665. https://pubs.acs.org/doi/full/10.1021/acs.chemrev.3c00667 (Mar. 2024).

19. NREL. *Interactive Best Research-Cell Efficiency Chart | Photovoltaic Research | NREL* 2024. https://www.nrel.gov/pv/interactive-cell-efficiency.html.

20. Mazumdar, S., Zhao, Y. & Zhang, X. Stability of Perovskite Solar Cells: Degradation Mechanisms and Remedies. *Frontiers in Electronics* **2,** 712785. ISSN: 2673-5857 (Aug. 2021).

21. McGovern, L., Alarcón-Lladó, E., Garnett, E. C., Ehrler, B. & van der Zwaan, B. Perovskite Solar Modules for the Residential Sector. *ACS Energy Letters* **8,** 4862–4866. ISSN: 23808195. http://pubs.acs.org/journal/aelccp (Nov. 2023).

22. Zhang, D., Li, D., Hu, Y., Mei, A. & Han, H. *Degradation pathways in perovskite solar cells and how to meet international standards* Dec. 2022.

23. Holzhey, P. & Saliba, M. A full overview of international standards assessing the long-term stability of perovskite solar cells † (2018).

24. Arndt, R. & Robert Puto, I. Basic Understanding of IEC Standard Testing For Photovoltaic Panels.

25. Das, C. *et al.* Unraveling the Role of Perovskite in Buried Interface Passivation. *ACS Applied Materials and Interfaces* **15,** 56500–56510. ISSN: 19448252. /pmc/articles/PMC10711719/%20/pmc/articles/PMC10711719/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10711719/ (Dec. 2023).

26. Wagner, L. *et al.* The resource demands of multi-terawatt-scale perovskite tandem photovoltaics. *Joule* **8,** 1142–1160. ISSN: 2542-4351 (Apr. 2024).

27. IEA. *Net Zero by 2050 – Analysis - IEA* May 2021. https://www.iea.org/reports/net-zero-by-2050.

28. Goetz, K. P., Taylor, A. D., Hofstetter, Y. J. & Vaynzof, Y. *Sustainability in Perovskite Solar Cells* Jan. 2021.

29. European Chemical Society. *Element Scarcity - EuChemS Periodic Table - EuChemS* 2023. https://www.euchems.eu/euchems-periodic-table/.

30. European Chemical Society. *Element Scarcity Table Support Notes* Sept. 2018. https://www.euchems.eu/wp-content/uploads/2018/09/IYPT2019_Support_notes.pdf.

31. United Nations. *Take Action for the Sustainable Development Goals - United Nations Sustainable Development* https://www.un.org/sustainabledevelopment/sustainable-development-goals/.

32. *Element scarcity displayed in new EuChemS Periodic Table of Elements - EuChemS* https://www.euchems.eu/element-scarcity-displayed-in-new-euchems-periodic-table-of-elements/.

33. Kim, G. Y. *et al. Sustainable and environmentally viable perovskite solar cells* Apr. 2023.

34. Li, J. *et al.* Biological impact of lead from halide perovskites reveals the risk of introducing a safe threshold. *Nature Communications 2020 11:1* **11,** 1–5. ISSN: 2041-1723. https://www.nature.com/articles/s41467-019-13910-y (Jan. 2020).

35. ECHA. *Cesium Lead Tribromide Substance Information* https://echa.europa.eu/substance-information/-/substanceinfo/100.271.566.

36. ECHA. *Lead* https://echa.europa.eu/hot-topics/lead.

37. Macdonald, T. J., Lanzetta, L., Liang, X., Ding, D. & Haque, S. A. *Engineering Stable Lead-Free Tin Halide Perovskite Solar Cells: Lessons from Materials Chemistry* June 2023.

38. Rencheck, M. L., Libby, C., Montgomery, A. & Stein, J. S. Managing potential environmental and human health risks of lead halide perovskite photovoltaic modules. http://creativecommons.org/licenses/by/4.0/ (2024).

39. Weyand, S., Kawajiri, K., Mortan, C., Zeller, V. & Schebek, L. Are Perovskite Solar Cells an Environmentally Sustainable Emerging Energy Technology? Upscaling from Lab to Fab in Life Cycle Assessment. *ACS Sustainable Chemistry and Engineering* **11,** 14010–14019. ISSN: 21680485 (Sept. 2023).

40. Aktas, E. *et al.* Challenges and strategies toward long-term stability of lead-free tin-based perovskite solar cells. *Communications Materials 2022 3:1* **3,** 1–14. ISSN: 2662-4443. https://www.nature.com/articles/s43246-022-00327-2 (Dec. 2022).

41. Patel, S. K., Nayak, S. & Senanayak, S. P. Impact of B-Site Cation Substitution on Ionic and Electronic Charge Transport in Metal Halide Perovskites. *ACS Applied Electronic Materials.* ISSN: 26376113 (Oct. 2023).

42. *Metal-Halide Perovskite Semiconductors* (Springer International Publishing, 2023).

43. Klug, M. T. *et al.* Metal composition influences optoelectronic quality in mixed-metal lead-tin triiodide perovskite solar absorbers. *Energy and Environmental Science* **13,** 1776–1787. ISSN: 17545706 (June 2020).

44. Turren-Cruz, S. H. *et al.* Multicomponent Approach for Stable Methylammonium-Free Tin-Lead Perovskite Solar Cells. *ACS Energy Letters,* 432–441. ISSN: 23808195. https://pubs.acs.org/doi/full/10.1021/acsenergylett.3c02426 (2024).

45. Bordovalos, A. *et al.* Implications of Electron Transport Layer and Back Metal Contact Variations in Tin-Lead Perovskite Solar Cells Assessed by Spectroscopic Ellipsometry and External Quantum Efficiency. *ACS Applied Materials and Interfaces* **15,** 19730–19740. ISSN: 19448252. www.acsami.org (Apr. 2023).

46. Eperon, G. E. *et al.* Perovskite-perovskite tandem photovoltaics with optimized band gaps. *Science* **354,** 861–865. ISSN: 10959203. https://www.science.org/doi/10.1126/science.aaf9717 (Nov. 2016).

47. Leijtens, T., Prasanna, R., Gold-Parker, A., Toney, M. F. & McGehee, M. D. Mechanism of Tin Oxidation and Stabilization by Lead Substitution in Tin Halide Perovskites. *ACS Energy Letters* **2,** 2159–2165. ISSN: 23808195. https://pubs.acs.org/doi/full/10.1021/acsenergylett.7b00636 (Sept. 2017).

48. Turren-Cruz, S. H., Hagfeldt, A. & Saliba, M. Methylammonium-free, high-performance, and stable perovskite solar cells on a planar architecture. *Science* **362,** 449–453. ISSN: 10959203. https://www.science.org/doi/10.1126/science.aat3583 (Oct. 2018).

49. Imran, T. *et al.* Methylammonium and Bromide-Free Tin-Based Low Bandgap Perovskite Solar Cells. *Advanced Energy Materials* **12,** 2200305. ISSN: 1614-6840. https://onlinelibrary.wiley.com/doi/full/10.1002/aenm.202200305%20https://onlinelibrary.wiley.com/doi/abs/10.1002/aenm.202200305%20https://onlinelibrary.wiley.com/doi/10.1002/aenm.202200305 (July 2022).

50. Prasanna, R. *et al.* Band Gap Tuning via Lattice Contraction and Octahedral Tilting in Perovskite Materials for Photovoltaics. *Journal of the American Chemical Society* **139,** 11117–11124. ISSN: 15205126. https://pubs.acs.org/doi/full/10.1021/jacs.7b04981 (Aug. 2017).

51. Leijtens, T. *et al.* Tin–lead halide perovskites with improved thermal and air stability for efficient all-perovskite tandem solar cells. *Sustainable Energy & Fuels* **2,** 2450–2459. ISSN: 23984902. https://pubs.rsc.org/en/content/articlehtml/2018/se/c8se00314a%20https://pubs.rsc.org/en/content/articlelanding/2018/se/c8se00314a (Oct. 2018).

52. Shao, S. & Loi, M. A. *The Role of the Interfaces in Perovskite Solar Cells* Jan. 2020.

53. Manspeaker, C., Venkatesan, S., Zakhidov, A. & Martirosyan, K. S. *Role of interface in stability of perovskite solar cells* Feb. 2017.

54. Marchioro, A. *et al.* Unravelling the mechanism of photoinduced charge transfer processes in lead iodide perovskite solar cells. *Nature Photonics* **8,** 250–255. ISSN: 17494885 (Mar. 2014).

55. Deepika, Singh, A., Verma, U. K. & Tonk, A. Device Structures of Perovskite Solar Cells: A Critical Review. *Physica Status Solidi (A) Applications and Materials Science* **220.** ISSN: 18626319 (May 2023).

56. Warby, J. *et al.* Understanding Performance Limiting Interfacial Recombination in pin Perovskite Solar Cells. *Advanced Energy Materials* **12.** ISSN: 16146840 (Mar. 2022).

57. Huerta Hernandez, L. *et al. Factors Limiting the Operational Stability of Tin-Lead Perovskite Solar Cells* Jan. 2023.

58. Lim, V. J. *et al.* Air-Degradation Mechanisms in Mixed Lead-Tin Halide Perovskites for Solar Cells. *Advanced Energy Materials* **13.** ISSN: 16146840 (Sept. 2023).

59. Rombach, F. *et al.* Disentangling Degradation Pathways of Narrow Bandgap Lead-Tin Perovskite Material and Photovoltaic Devices. https://www.researchsquare.com%20https://www.researchsquare.com/article/rs-4502930/v1 (June 2024).

60. Huerta Hernandez, L. *et al.* The role of A-site composition in the photostability of tin-lead perovskite solar cells. *Sustainable Energy and Fuels* **6,** 4605–4613. ISSN: 23984902 (Sept. 2022).

61. Yu, W. *et al.* Recent advances on interface engineering of perovskite solar cells. ISSN: 1998-0124. https://doi.org/10.1007/s12274-021-3488-7 (2020).

62. Lim, K.-G. *et al.* Effect of Interfacial Layers on the Device Lifetime of Perovskite Solar Cells. *Small Methods* **4,** 2000065. ISSN: 2366-9608. https://onlinelibrary.wiley.com/doi/full/10.1002/smtd.202000065%20https://onlinelibrary.wiley.com/doi/abs/10.1002/smtd.202000065%20https://onlinelibrary.wiley.com/doi/10.1002/smtd.202000065 (Aug. 2020).

63. Wang, D., Ye, T. & Zhang, Y. Recent advances of non-fullerene organic electron transport materials in perovskite solar cells. *Journal of Materials Chemistry A* **8,** 20819–20848. ISSN: 20507496 (Oct. 2020).

64. Hu, S., Smith, J. A., Snaith, H. J. & Wakamiya, A. Prospects for Tin-Containing Halide Perovskite Photovoltaics. *Precision Chemistry* **1,** 69–82. ISSN: 27719316. https://pubs.acs.org/doi/full/10.1021/prechem.3c00018 (Apr. 2023).

65. Gao, H. *et al.* Thermally Stable All-Perovskite Tandem Solar Cells Fully Using Metal Oxide Charge Transport Layers and Tunnel Junction. *Solar RRL* **5,** 2100814. ISSN: 2367-198X. https://onlinelibrary.wiley.com/doi/full/10.1002/solr.202100814%20https://onlinelibrary.wiley.com/doi/abs/10.1002/solr.202100814%20https://onlinelibrary.wiley.com/doi/10.1002/solr.202100814 (Dec. 2021).

66. Li, M. *et al.* Interface Modification by Ionic Liquid: A Promising Candidate for Indoor Light Harvesting and Stability Improvement of Planar Perovskite Solar Cells. *Advanced Energy Materials* **8.** ISSN: 16146840 (Aug. 2018).

67. Lee, H., Kang, S. B., Lee, S., Zhu, K. & Kim, D. H. Progress and outlook of Sn–Pb mixed perovskite solar cells. *Nano Convergence* **10,** 1–16. ISSN: 21965404. https://link.springer.com/articles/10.1186/s40580-023-00371-9%20https://link.springer.com/article/10.1186/s40580-023-00371-9 (Dec. 2023).

68. Wijesekara, A. *et al.* Enhanced Stability of Tin Halide Perovskite Photovoltaics Using a Bathocuproine—Copper Top Electrode. *Advanced Energy Materials* **11,** 2102766. ISSN: 1614-6840. https://onlinelibrary.wiley.com/doi/full/10.1002/aenm.202102766%20https://onlinelibrary.wiley.com/doi/abs/10.1002/aenm.202102766%20https://onlinelibrary.wiley.com/doi/10.1002/aenm.202102766 (Dec. 2021).

69. Dey, K. *et al.* Substitution of lead with tin suppresses ionic transport in halide perovskite optoelectronics. *Energy & Environmental Science* **17,** 760–769. ISSN: 17545706. https://pubs.rsc.org/en/content/articlehtml/2024/ee/d3ee03772j%20https://pubs.rsc.org/en/content/articlelanding/2024/ee/d3ee03772j (Jan. 2024).

70. Dey, K. *et al.* Supporting Information Substitution of Lead with Tin Suppresses Ionic Transport in Halide Perovskite Optoelectronics (2023).

71. Hu, S. *et al.* Optimized carrier extraction at interfaces for 23.6% efficient tin-lead perovskite solar cells. *Energy and Environmental Science.* ISSN: 17545706 (2022).

72. Li, H. *et al.* Surface Dedoping for Methylammonium-Free Tin-Lead Perovskite Solar Cells. *ACS Energy Letters* **9,** 400–409. ISSN: 23808195. https://pubs.acs.org/doi/full/10.1021/acsenergylett.3c01682 (Feb. 2024).

73. Zhang, J. *et al.* A Fully Robotic Platform for Optimizing the High-Dimensional Processing Parameter Space of Perovskite Thin-Films. *SSRN Electronic Journal.* https://papers.ssrn.com/abstract=4309089 (Dec. 2022).

74. Yu, Y. *et al.* Improving the Performance of Formamidinium and Cesium Lead Triiodide Perovskite Solar Cells using Lead Thiocyanate Additives. *ChemSusChem* **9,** 3288–3297. ISSN: 1864564X (Dec. 2016).

75. Kamaraki, C. *et al.* Charting the Irreversible Degradation Modes of Low Bandgap Pb-Sn Perovskite Compositions for De-Risking Practical Industrial Development. *Advanced Energy Materials.* ISSN: 16146840 (Mar. 2024).

76. Liao, W. *et al.* Lead-Free Inverted Planar Formamidinium Tin Triiodide Perovskite Solar Cells Achieving Power Conversion Efficiencies up to 6.22%. *Advanced Materials* **28,** 9333–9340. ISSN: 1521-4095. https://onlinelibrary.wiley.com/doi/full/10.1002/adma.201602992%20https://onlinelibrary.wiley.com/doi/abs/10.1002/adma.201602992%20https://onlinelibrary.wiley.com/doi/10.1002/adma.201602992 (Nov. 2016).

77. Kapil, G. *et al.* Strain Relaxation and Light Management in Tin-Lead Perovskite Solar Cells to Achieve High Efficiencies. *ACS Energy Letters* **4,** 1991–1998. ISSN: 23808195. https://pubs.acs.org/doi/full/10.1021/acsenergylett.9b01237 (Aug. 2019).

78. Kurisinkal Pious, J. *et al.* Revealing the Role of Tin Fluoride Additive in Narrow Bandgap Pb-Sn Perovskites for Highly Efficient Flexible All-Perovskite Tandem Cells. *ACS Applied Materials and Interfaces* **15,** 10150–10157. ISSN: 19448252. https://pubs.acs.org/doi/full/10.1021/acsami.2c19124 (2022).

79. Ho, K., Wei, M., Sargent, E. H. & Walker, G. C. Grain Transformation and Degradation Mechanism of Formamidinium and Cesium Lead Iodide Perovskite under Humidity and Light. *ACS Energy Letters* **6,** 934–940. ISSN: 23808195. https://pubs.acs.org/doi/full/10.1021/acsenergylett.0c02247 (Mar. 2021).

80. Nielsen, C. B., Holliday, S., Chen, H. Y., Cryer, S. J. & McCulloch, I. Non-Fullerene Electron Acceptors for Use in Organic Solar Cells. *Accounts of Chemical Research* **48,** 2803–2812. ISSN: 15204898. https://pubs.acs.org/doi/full/10.1021/acs.accounts.5b00199 (Oct. 2015).

81. Bastos, J. P. *et al.* Oxygen-Induced Degradation in C60-Based Organic Solar Cells: Relation between Film Properties and Device Performance. *ACS Applied Materials and Interfaces* **8,** 9798–9805. ISSN: 19448252. www.acsami.org (May 2016).

82. Nisha, J. A. *et al.* X-ray diffraction and thermoanalytical investigations of amorphous carbons derived from C60. *Thermochimica Acta* **286,** 17–24. ISSN: 0040-6031 (Sept. 1996).

Appendix

A IEA Roadmap

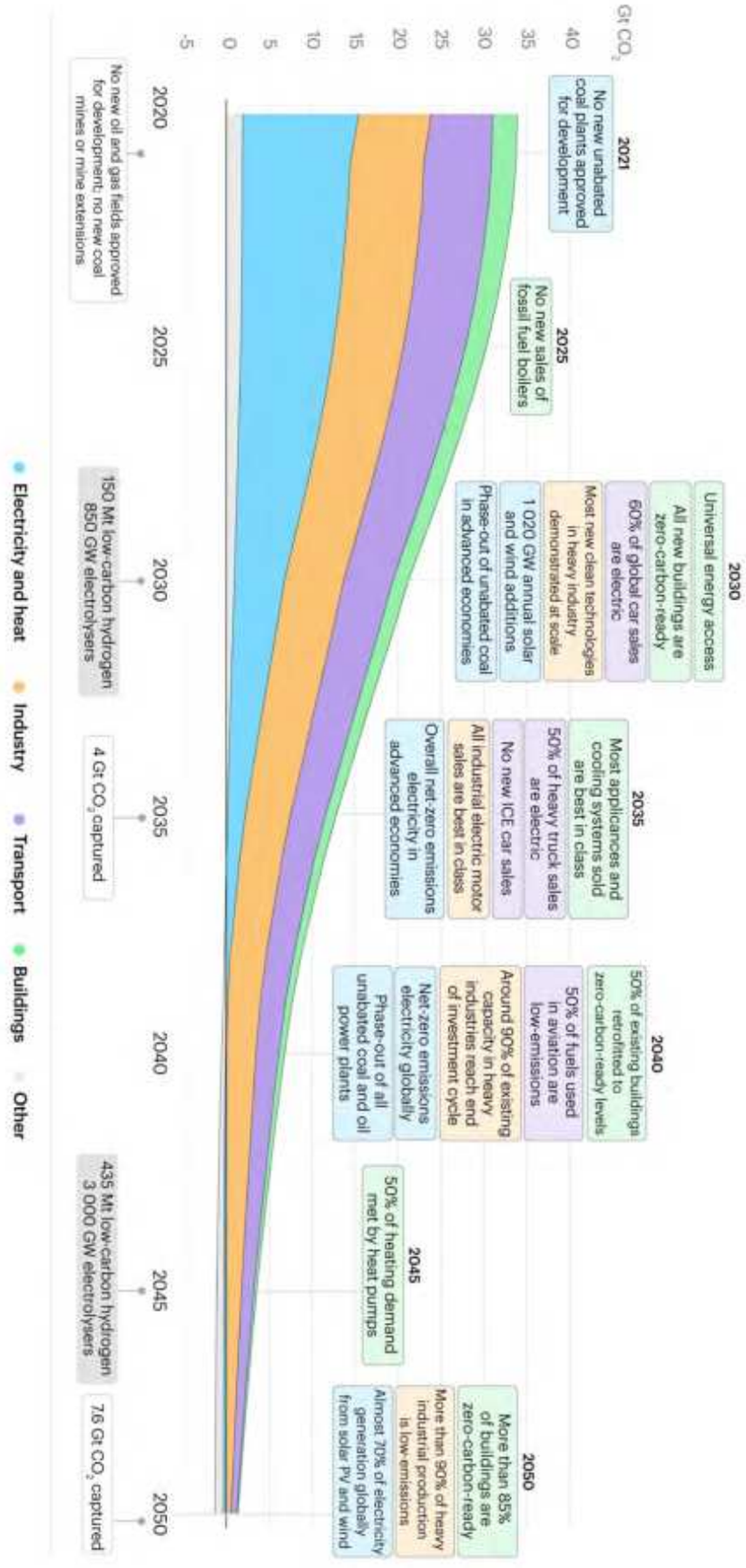


Figure 1: Adapted from the IEA Net Zero by 2050 Roadmap for the Global Energy Sector. [27]

B ITO Substrate Pattern

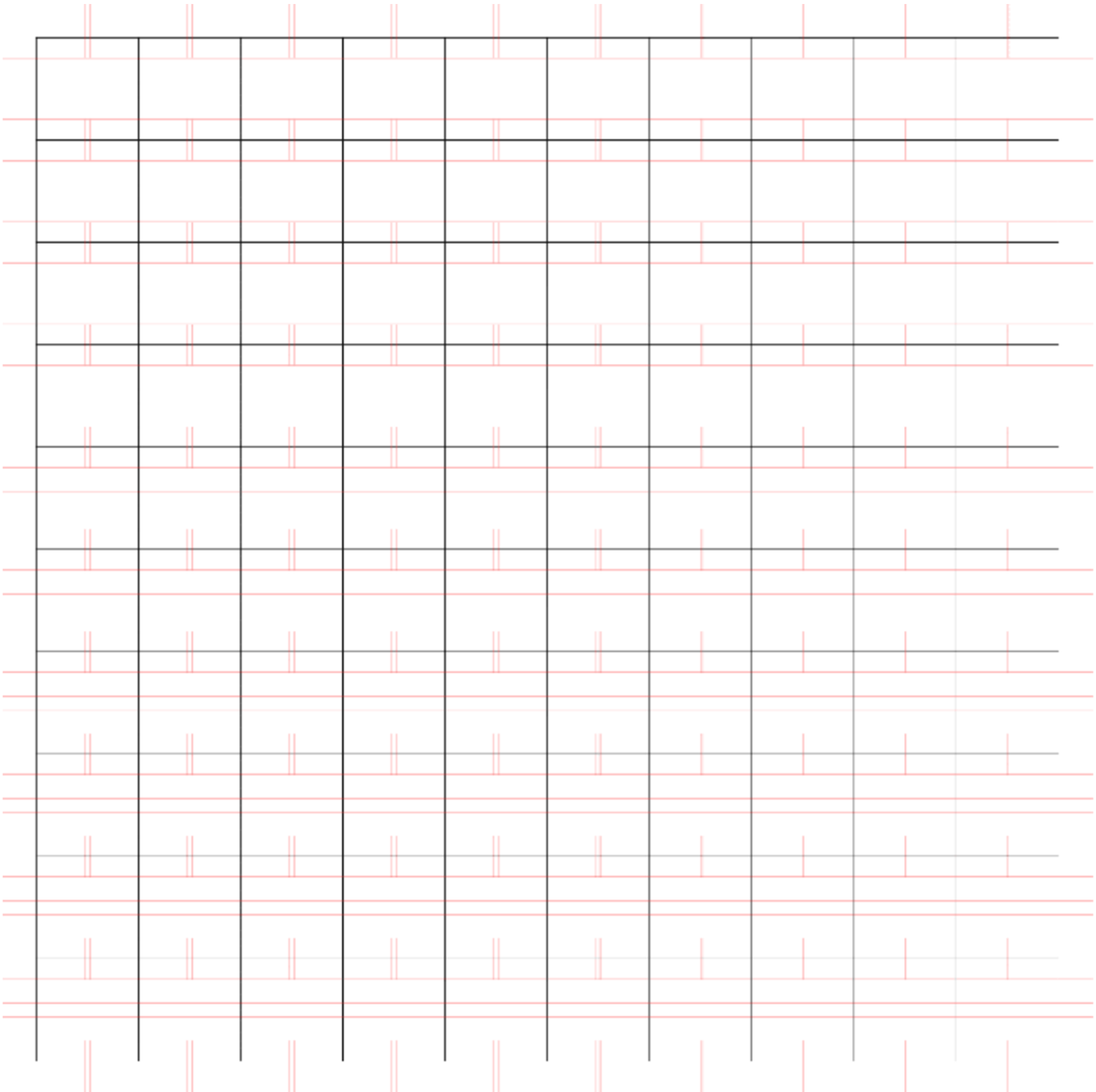


Figure 2: Pattern for ITO on glass substrates.

C Aged Devices

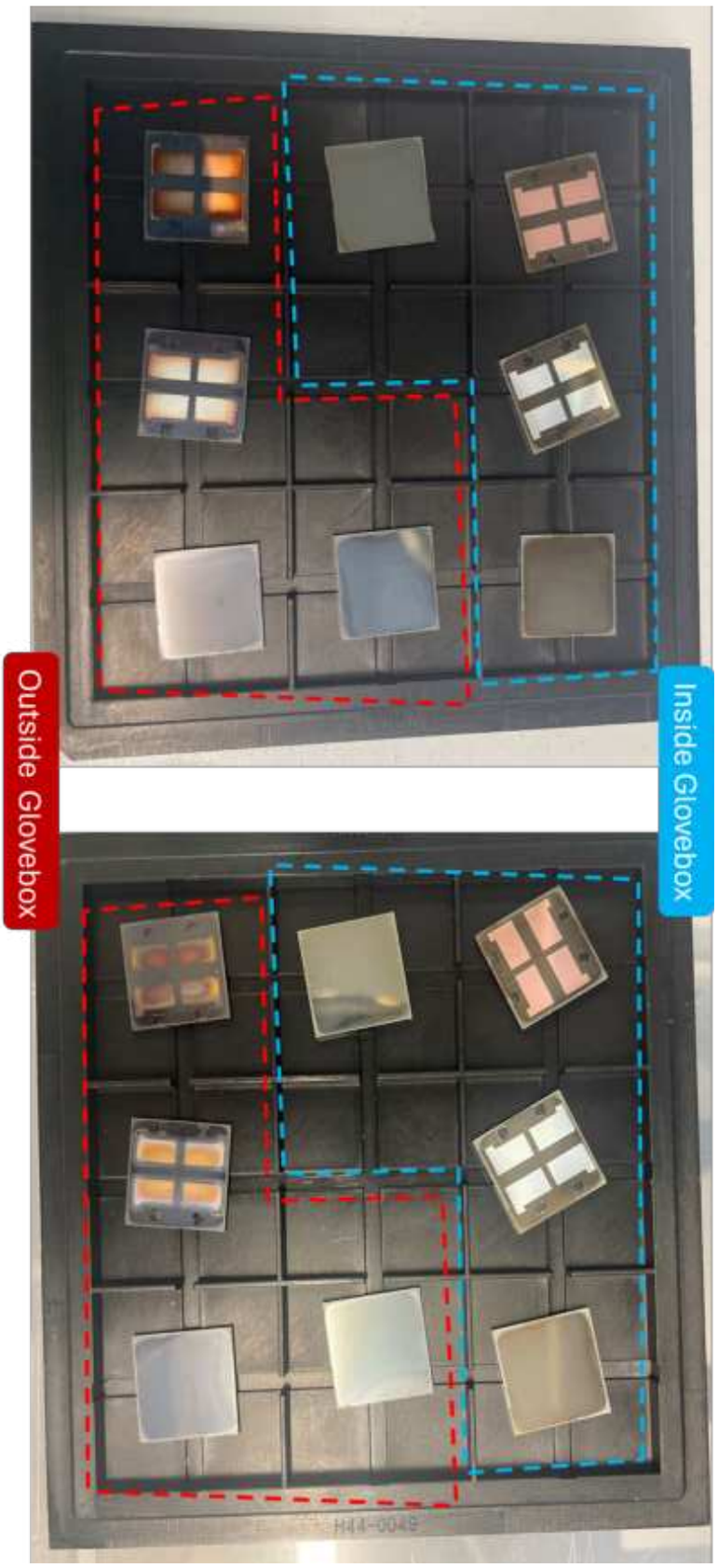


Figure 3: Photo of the aged samples, after 1 week (left) and 3 weeks (right).

D Daily UV-Vis Measurments

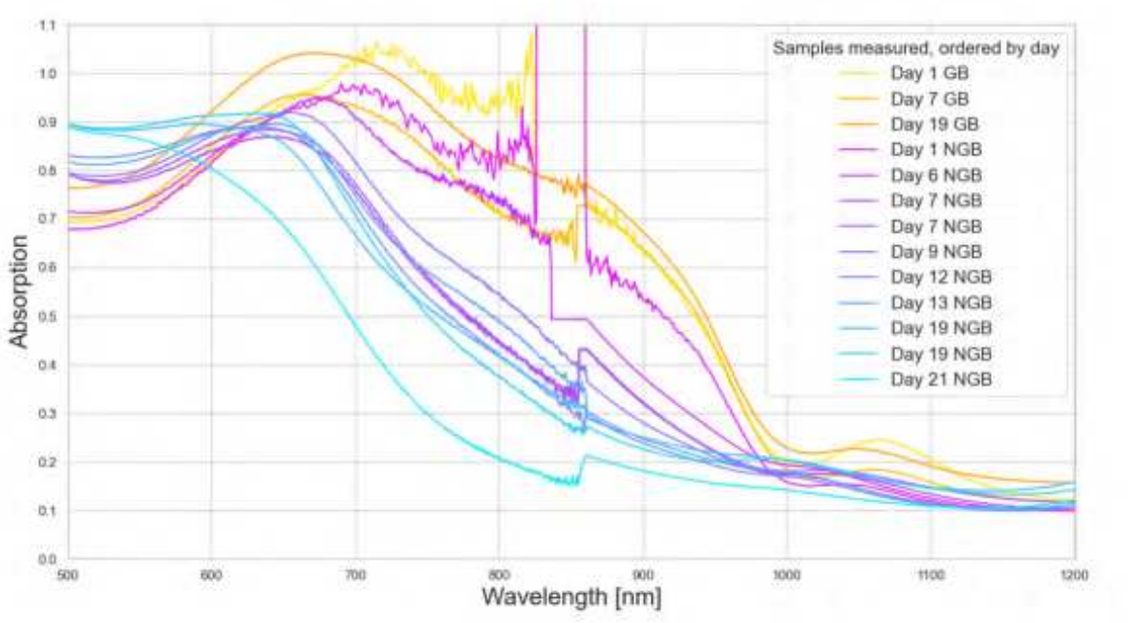


Figure 4: Daily absorbance measurements for Copper

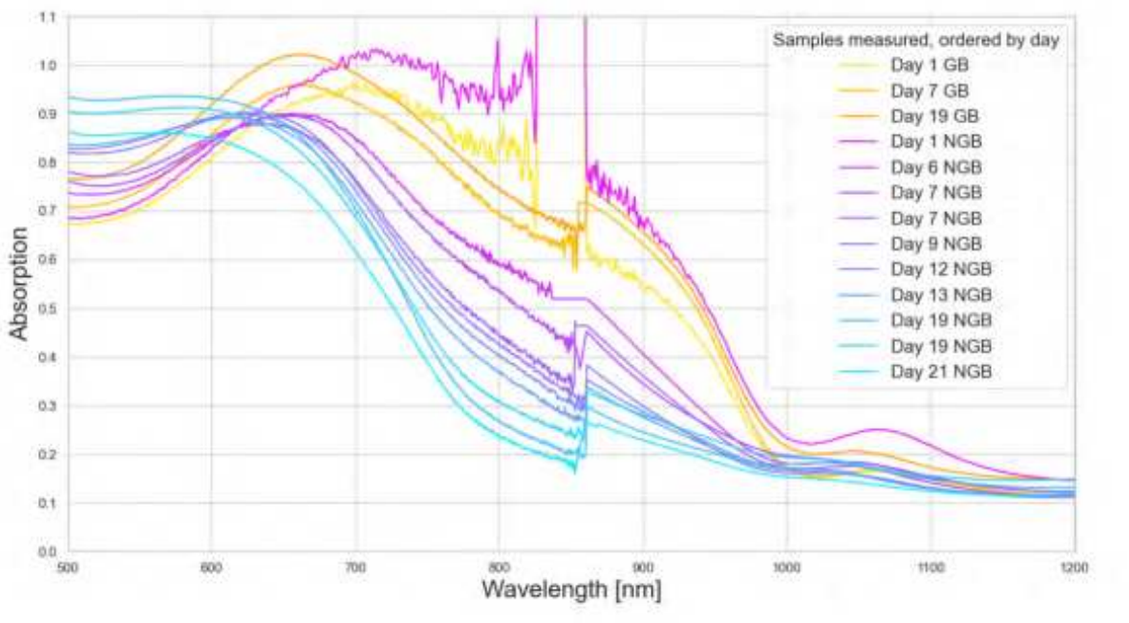


Figure 5: Daily absorbance measurements for Silver

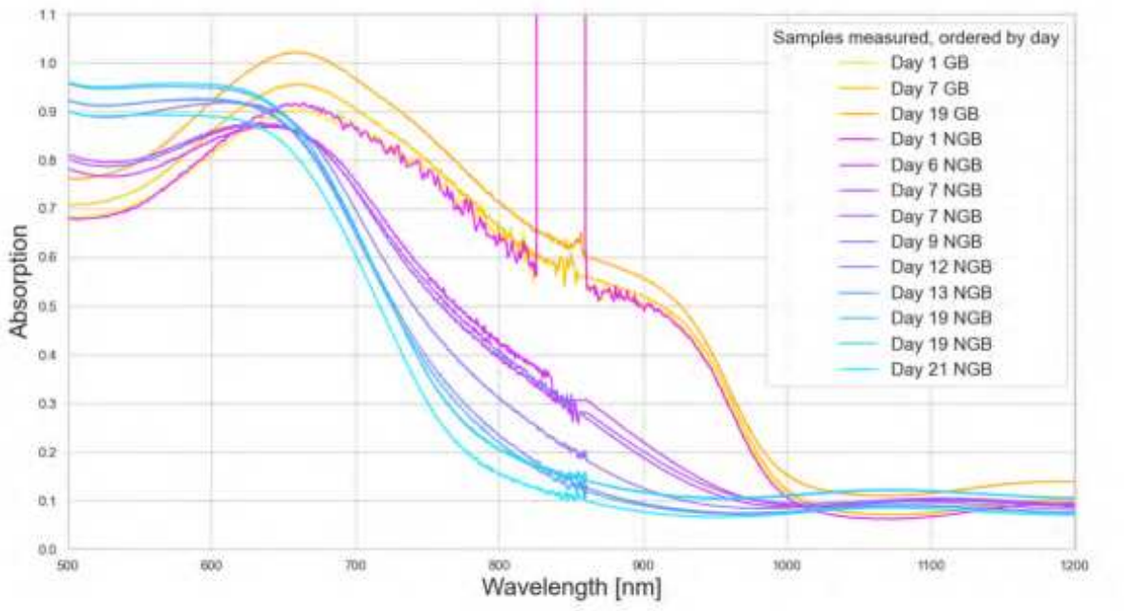


Figure 6: Daily absorbance measurements for No Metal

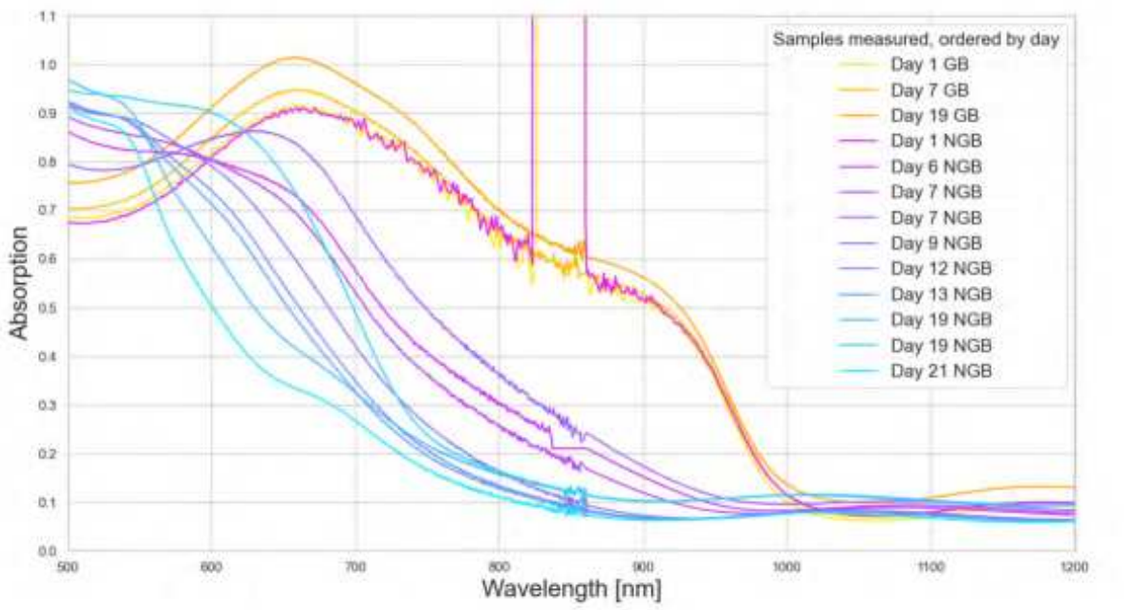


Figure 7: Daily absorbance measurements for No ETL

E EDX of SnF_2

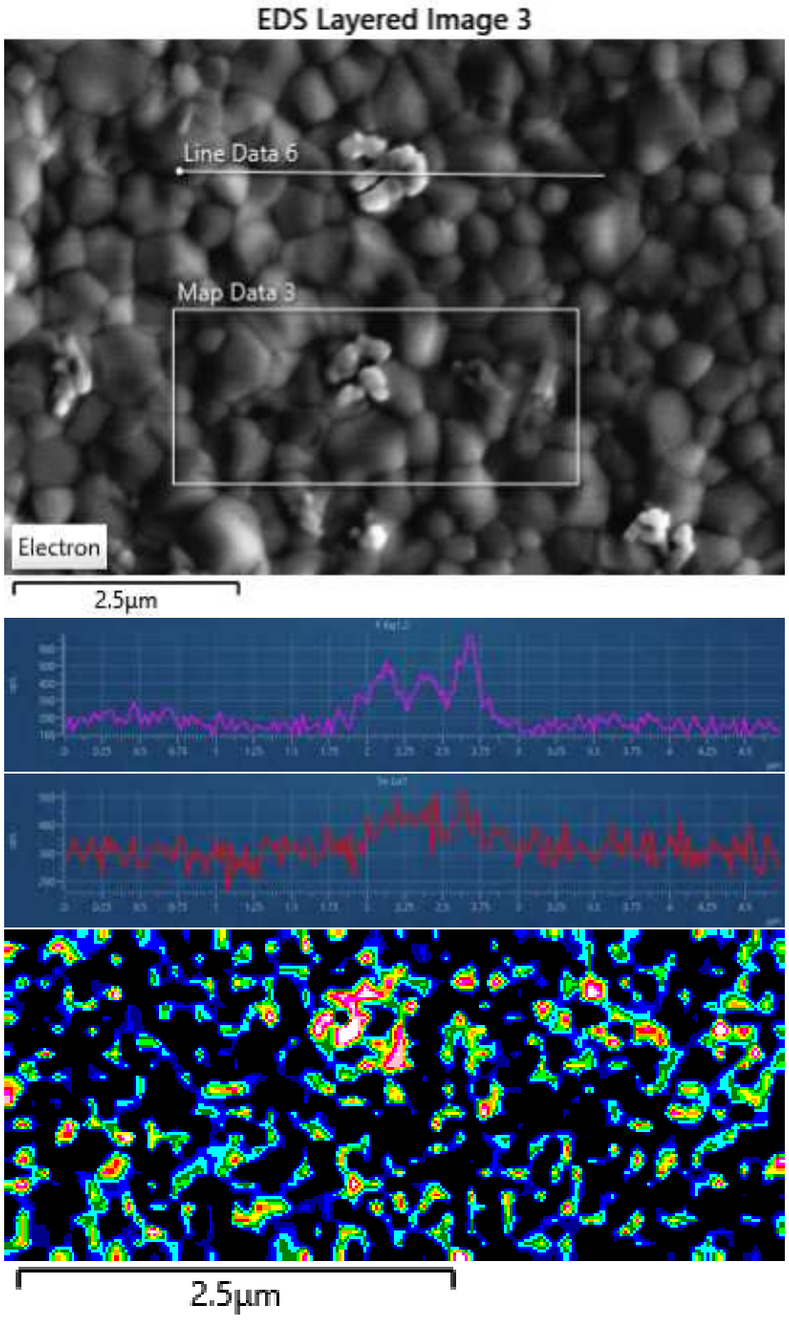


Figure 8: SnF_2 identification in EDX maps and linescan

F Beam Damage

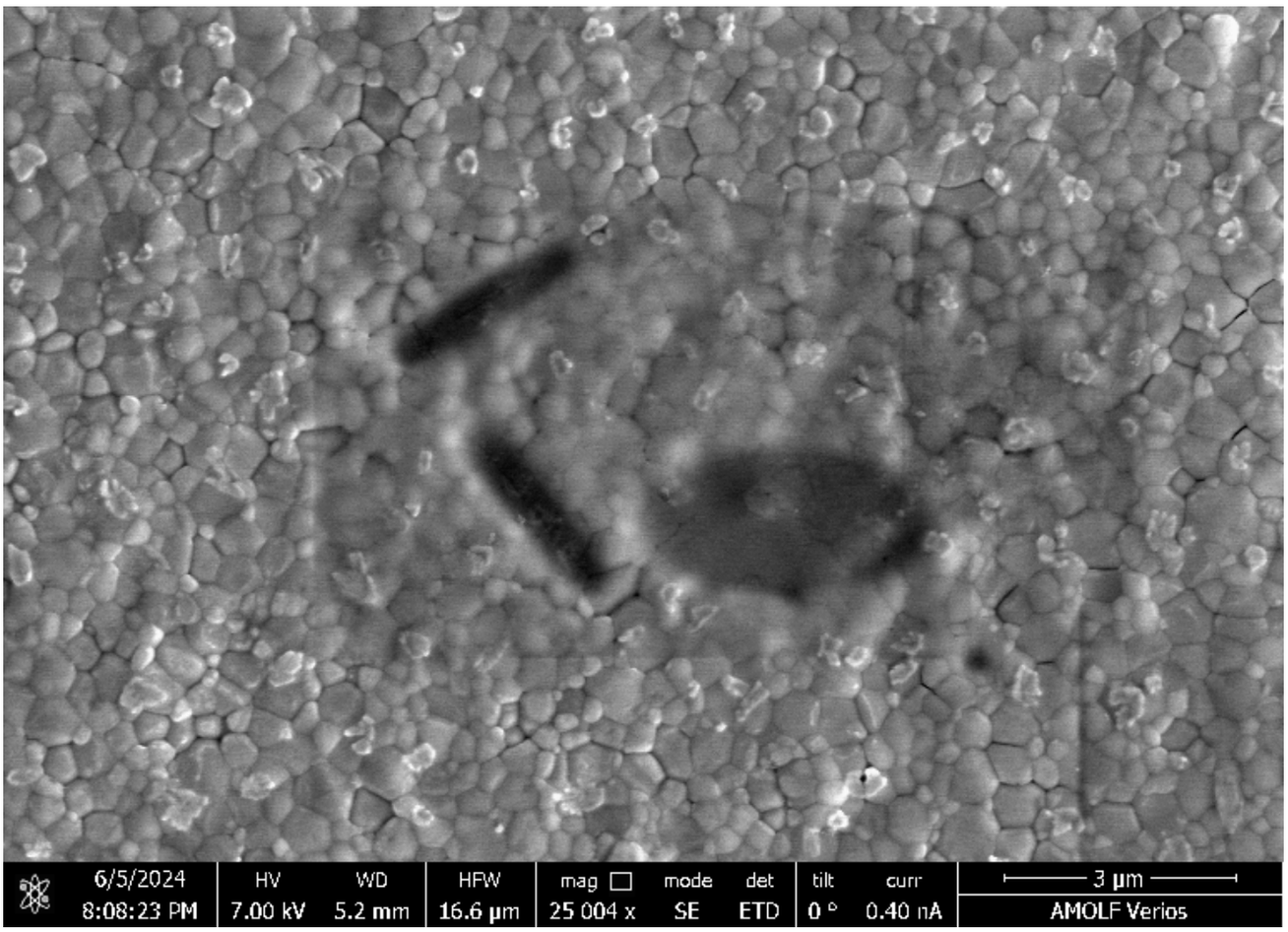


Figure 9: Beam damage from running linescans and maps during EDX analysis.

G Light-Soaking Code

These light-soaking Spyder files were written by Moritz C. Schmidt and Daphne M. Dekker.

automated_light_soaking_script_fiona.py

```
    from pathlib import Path
import sys
import time
import os
import numpy as np

sys.path.insert(
    0, r'C:\Users\Paios\Documents\Data\Fiona\Light soaking tools\agilentCharacterization')

sys.path.insert(0, r'C:\Users\Paios\Documents\Data\Fiona\Light soaking tools\Pixelswitch')

sys.path.insert(0, r'C:\Users\Paios\Documents\Data\Fiona\Light soaking tools\Pico')

if True:
    from Pixel import Pixel
    from Keithley2400 import Keithley2400
    from pixelswitch import switch_pixel
    from Pico import Pico


keithley = Keithley2400(source='GPIB0::24::INSTR')

pico = Pico(ip='192.87.154.137', id='00000000565895fd')

experimentname = 'FM017_Dark_Voc'

device = 'd08'

pixels = []

pixels.append(Pixel(connection='A1',
                    pixel='P1',
                    name='{}_{}_{}_{}_P1'.format({},device,{},{}),
                    path=Path(
                        r'C:\Users\Paios\Documents\Data\Fiona\20240531_FM017'),
                    active_area=0.04))
pixels.append(Pixel(connection='A2',
                    pixel='P2',
                    name='{}_{}_{}_{}_P2'.format({},device,{},{}),
                    path=Path(
                        r'C:\Users\Paios\Documents\Data\Fiona\20240531_FM017'),
                    active_area=0.04))
pixels.append(Pixel(connection='A3',
                    pixel='P3',
                    name='{}_{}_{}_{}_P3'.format({},device,{},{}),
                    path=Path(
                        r'C:\Users\Paios\Documents\Data\Fiona\20240531_FM017'),
                    active_area=0.04))
pixels.append(Pixel(connection='A4',
                    pixel='P4',
                    name='{}_{}_{}_{}_P4'.format({},device,{},{}),
                    path=Path(
                        r'C:\Users\Paios\Documents\Data\Fiona\20240531_FM017'),
                    active_area=0.04))


```

```
led = 'Pico'
pico.load_spectrum(r'C:\Users\Paios\Documents\Data\Fiona\Light soaking tools\Pico\spectrumAM1p5g.json')

pico_intensity = [100] # Intensity in %


t_lightsoaking = [0] + [600]*12 + [3600]*10 # in s (JV curve every 10 min for 2 h, then once every hour for 10 more hours, 12 h total)
light_soaking_iterations = len(t_lightsoaking)

t_recovery = [300]*4 + [600]*4 + [3600]*11 # in s (JV curve after 5, 10, 15 and 20 min, then after 30, 40, 50, 60 minutes, then once
     every hour for 11 h)
recovery_iterations = len(t_recovery)

for pixel in pixels:
    print(pixel.name)
    temppath = Path(pixel.path)

    if not os.path.exists(temppath):
        os.makedirs(temppath)
    pixel.temppath = temppath

#%% Single dark curve
illum = 'dark'
t = '00000_2'
folder = 'Dark_Voc'

#pixel = pixels[0]

for pixel in pixels:
    switch_pixel(pixel.connection)

    keithley.sweep_jv(output=Path(pixel.temppath),
                experimentname=experimentname,
                devicename=pixel.name.format(folder,illum,t),
                led=led,
                t_lightsoaking=0,
                led_current=100,
                temperature='RT',
                start_voltage=-0.2,
                end_voltage=1.2,
                voltage_step=0.04,
                scan_speed=0.1,
                direction='bidirectional',
                prebiasV=0,
                prebiasT=0,
                active_area=pixel.active_area,
                save_plots=True)

#%% Single light curve
illum = 'light'
t = '00000_check'
folder = 'Dark_Voc'

pico.turn_on()

#pixel = pixels[2]
for pixel in pixels:
    switch_pixel(pixel.connection)

    keithley.sweep_jv(output=Path(pixel.temppath),
              experimentname=experimentname,
              devicename=pixel.name.format(folder,illum,t),
              led=led,
              t_lightsoaking=0,
              led_current=100,
```

```
            temperature='RT',
            start_voltage=-0.2,
            end_voltage=1.2,
            voltage_step=0.04,
            scan_speed=0.1,
            direction='bidirectional',
            prebiasV=0,
            prebiasT=0,
            active_area=pixel.active_area,
            save_plots=True)

pico.turn_off()

#%% Long term light soaking at Voc + automated recovery

illum = 'light'
folder = 'Light_Voc'

pico.turn_on()

for i in range(light_soaking_iterations):

    time.sleep(t_lightsoaking[i])

    for pixel in pixels:
        switch_pixel(pixel.connection)

        keithley.sweep_jv(output=Path(pixel.temppath),
                        experimentname=experimentname,
                        devicename=pixel.name.format(folder,illum,str(sum(t_lightsoaking[:i+1])).zfill(5)),
                        led=led,
                        t_lightsoaking=sum(t_lightsoaking[:i+1]),
                        led_current=100,
                        temperature='RT',
                        start_voltage=-0.2,
                        end_voltage=1.2,
                        voltage_step=0.04,
                        scan_speed=0.1,
                        direction='bidirectional',
                        prebiasV=0,
                        prebiasT=0,
                        active_area=pixel.active_area,
                        save_plots=False)

pico.turn_off()

illum = 'dark'
t = str(sum(t_lightsoaking)).zfill(5)

for pixel in pixels:
    switch_pixel(pixel.connection)

    keithley.sweep_jv(output=Path(pixel.temppath),
                experimentname=experimentname,
                devicename=pixel.name.format(folder,illum,t),
                led=led,
                t_lightsoaking=sum(t_lightsoaking),
                led_current=100,
                temperature='RT',
                start_voltage=-0.2,
                end_voltage=1.2,
                voltage_step=0.04,
                scan_speed=0.1,
                direction='bidirectional',
                prebiasV=0,
```

```
                prebiasT=0,
                active_area=pixel.active_area,
                save_plots=False)

illum = 'light'
folder = 'Light_Voc_recovery'
for j in range(recovery_iterations):

    time.sleep(t_recovery[j])

    pico.turn_on()

    for pixel in pixels:
        switch_pixel(pixel.connection)

        keithley.sweep_jv(output=Path(pixel.temppath),
                        experimentname=experimentname,
                        devicename=pixel.name.format(folder,illum,str(sum(t_recovery[:j+1])).zfill(5)),
                        led=led,
                        t_lightsoaking=sum(t_recovery[:j+1]),
                        led_current=100,
                        temperature='RT',
                        start_voltage=-0.2,
                        end_voltage=1.2,
                        voltage_step=0.04,
                        scan_speed=0.1,
                        direction='bidirectional',
                        prebiasV=0,
                        prebiasT=0,
                        active_area=pixel.active_area,
                        save_plots=False)

    pico.turn_off()

illum = 'dark'
t = '{}'.format(str(sum(t_recovery)).zfill(5))
folder = 'Light_Voc_recovery'

keithley.sweep_jv(output=Path(pixel.temppath),
             experimentname=experimentname,
             devicename=pixel.name.format(folder,illum,t),
             led=led,
             t_lightsoaking=0,
             led_current=100,
             temperature='RT',
             start_voltage=-0.2,
             end_voltage=1.2,
             voltage_step=0.04,
             scan_speed=0.1,
             direction='bidirectional',
             prebiasV=0,
             prebiasT=0,
             active_area=pixel.active_area,
             save_plots=False)

#%% Long term light soaking at 0V + automated recovery

illum = 'light'
folder = 'Light_0V'
px = 3

pico.turn_on()

pixel = pixels[px]
switch_pixel(pixel.connection)
```

```
251
252 for i in range(light_soaking_iterations):
253     if i == 0:
254         keithley.sweep_jv(output=Path(pixel.temppath),
255                           experimentname=experimentname,
256                           devicename=pixel.name.format(folder,illum,str(sum(t_lightsoaking[:i+1])).zfill(5)),
257                           led=led,
258                           t_lightsoaking=sum(t_lightsoaking[:i+1]),
259                           led_current=100,
260                           temperature='RT',
261                           start_voltage=-0.2,
262                           end_voltage=1.2,
263                           voltage_step=0.04,
264                           scan_speed=0.1,
265                           direction='bidirectional',
266                           prebiasV=0,
267                           prebiasT=0,
268                           active_area=pixel.active_area,
269                           save_plots=False)
270     else:
271         keithley.sweep_jv(output=Path(pixel.temppath),
272                           experimentname=experimentname,
273                           devicename=pixel.name.format(folder,illum,str(sum(t_lightsoaking[:i+1])).zfill(5)),
274                           led=led,
275                           t_lightsoaking=sum(t_lightsoaking[:i+1]),
276                           led_current=100,
277                           temperature='RT',
278                           start_voltage=-0.2,
279                           end_voltage=1.2,
280                           voltage_step=0.04,
281                           scan_speed=0.1,
282                           direction='bidirectional',
283                           prebiasV=0,
284                           prebiasT=t_lightsoaking[i],
285                           active_area=pixel.active_area,
286                           save_plots=False)
287
288 pico.turn_off()
289
290 illum = 'dark'
291 t = str(sum(t_lightsoaking)).zfill(5)
292
293 keithley.sweep_jv(output=Path(pixel.temppath),
294             experimentname=experimentname,
295             devicename=pixel.name.format(folder,illum,t),
296             led=led,
297             t_lightsoaking=sum(t_lightsoaking),
298             led_current=100,
299             temperature='RT',
300             start_voltage=-0.2,
301             end_voltage=1.2,
302             voltage_step=0.04,
303             scan_speed=0.1,
304             direction='bidirectional',
305             prebiasV=0,
306             prebiasT=0,
307             active_area=pixel.active_area,
308             save_plots=False)
309
310 illum = 'light'
311 folder = 'Light_OV_recovery'
312 for j in range(recovery_iterations):
313
314     time.sleep(t_recovery[j])
315
```

```
    pico.turn_on()

    keithley.sweep_jv(output=Path(pixel.temppath),
                      experimentname=experimentname,
                      devicename=pixel.name.format(folder,illum,str(sum(t_recovery[:j+1])).zfill(5)),
                      led=led,
                      t_lightsoaking=sum(t_recovery[:j+1]),
                      led_current=100,
                      temperature='RT',
                      start_voltage=-0.2,
                      end_voltage=1.2,
                      voltage_step=0.04,
                      scan_speed=0.1,
                      direction='bidirectional',
                      prebiasV=0,
                      prebiasT=0,
                      active_area=pixel.active_area,
                      save_plots=False)

    pico.turn_off()

illum = 'dark'
t = '{}'.format(str(sum(t_recovery)).zfill(5))
folder = 'Light_0V_recovery'

keithley.sweep_jv(output=Path(pixel.temppath),
             experimentname=experimentname,
             devicename=pixel.name.format(folder,illum,t),
             led=led,
             t_lightsoaking=0,
             led_current=100,
             temperature='RT',
             start_voltage=-0.2,
             end_voltage=1.2,
             voltage_step=0.04,
             scan_speed=0.1,
             direction='bidirectional',
             prebiasV=0,
             prebiasT=0,
             active_area=pixel.active_area,
             save_plots=False)

#%% Long term dark 0V + automated recovery

illum = 'light'
folder = 'Dark_0V'

for i in range(light_soaking_iterations):

    time.sleep(t_lightsoaking[i])

    pico.turn_on()

    for pixel in pixels:
        switch_pixel(pixel.connection)

        keithley.sweep_jv(output=Path(pixel.temppath),
                          experimentname=experimentname,
                          devicename=pixel.name.format(folder,illum,str(sum(t_lightsoaking[:i+1])).zfill(5)),
                          led=led,
                          t_lightsoaking=sum(t_lightsoaking[:i+1]),
                          led_current=100,
                          temperature='RT',
                          start_voltage=-0.2,
                          end_voltage=1.2,
```

```
                        voltage_step=0.04,
                        scan_speed=0.1,
                        direction='bidirectional',
                        prebiasV=0,
                        prebiasT=0,
                        active_area=pixel.active_area,
                        save_plots=False)

    pico.turn_off()

illum = 'dark'
t = str(sum(t_lightsoaking)).zfill(5)

for pixel in pixels:
    switch_pixel(pixel.connection)

    keithley.sweep_jv(output=Path(pixel.temppath),
                experimentname=experimentname,
                devicename=pixel.name.format(folder,illum,t),
                led=led,
                t_lightsoaking=sum(t_lightsoaking),
                led_current=100,
                temperature='RT',
                start_voltage=-0.2,
                end_voltage=1.2,
                voltage_step=0.04,
                scan_speed=0.1,
                direction='bidirectional',
                prebiasV=0,
                prebiasT=0,
                active_area=pixel.active_area,
                save_plots=False)

illum = 'light'
folder = 'Dark_0V_recovery'
for j in range(recovery_iterations):

    time.sleep(t_recovery[j])

    pico.turn_on()

    for pixel in pixels:
        switch_pixel(pixel.connection)


        keithley.sweep_jv(output=Path(pixel.temppath),
                        experimentname=experimentname,
                        devicename=pixel.name.format(folder,illum,str(sum(t_recovery[:j+1])).zfill(5)),
                        led=led,
                        t_lightsoaking=sum(t_recovery[:j+1]),
                        led_current=100,
                        temperature='RT',
                        start_voltage=-0.2,
                        end_voltage=1.2,
                        voltage_step=0.04,
                        scan_speed=0.1,
                        direction='bidirectional',
                        prebiasV=0,
                        prebiasT=0,
                        active_area=pixel.active_area,
                        save_plots=False)

    pico.turn_off()

illum = 'dark'
```

```
t = '{}'.format(str(sum(t_recovery)).zfill(5))
folder = 'Dark_0V_recovery'

for pixel in pixels:
    switch_pixel(pixel.connection)

    keithley.sweep_jv(output=Path(pixel.temppath),
                experimentname=experimentname,
                devicename=pixel.name.format(folder,illum,t),
                led=led,
                t_lightsoaking=0,
                led_current=100,
                temperature='RT',
                start_voltage=-0.2,
                end_voltage=1.2,
                voltage_step=0.04,
                scan_speed=0.1,
                direction='bidirectional',
                prebiasV=0,
                prebiasT=0,
                active_area=pixel.active_area,
                save_plots=False)

#%% Long term dark under bias + automated recovery
illum = 'light'
folder = 'Dark_Voc'
px = 3

pixel = pixels[px]
switch_pixel(pixel.connection)

for i in range(light_soaking_iterations):

    if i == 0:
        pico.turn_on()
        keithley.sweep_jv(output=Path(pixel.temppath),
                        experimentname=experimentname,
                        devicename=pixel.name.format(folder,illum,str(sum(t_lightsoaking[:i+1])).zfill(5)),
                        led=led,
                        t_lightsoaking=sum(t_lightsoaking[:i+1]),
                        led_current=100,
                        temperature='RT',
                        start_voltage=-0.2,
                        end_voltage=1.2,
                        voltage_step=0.04,
                        scan_speed=0.1,
                        direction='bidirectional',
                        prebiasV=0,
                        prebiasT=0,
                        active_area=pixel.active_area,
                        save_plots=False)
    else:
        keithley.sweep_jv(output=Path(pixel.temppath),
                        experimentname=experimentname,
                        devicename=pixel.name.format(folder,illum,str(sum(t_lightsoaking[:i+1])).zfill(5)),
                        led=led,
                        t_lightsoaking=sum(t_lightsoaking[:i+1]),
                        led_current=100,
                        temperature='RT',
                        start_voltage=-0.2,
                        end_voltage=1.2,
                        voltage_step=0.04,
                        scan_speed=0.1,
                        direction='bidirectional',
                        prebiasV=0.7,
```

```
                        prebiasT=t_lightsoaking[i],
                        active_area=pixel.active_area,
                        pico_after_bias=True,
                        save_plots=False)

    pico.turn_off()

illum = 'dark'
t = str(sum(t_lightsoaking)).zfill(5)

for pixel in pixels:
    switch_pixel(pixel.connection)

    keithley.sweep_jv(output=Path(pixel.temppath),
                experimentname=experimentname,
                devicename=pixel.name.format(folder,illum,t),
                led=led,
                t_lightsoaking=sum(t_lightsoaking),
                led_current=100,
                temperature='RT',
                start_voltage=-0.2,
                end_voltage=1.2,
                voltage_step=0.04,
                scan_speed=0.1,
                direction='bidirectional',
                prebiasV=0,
                prebiasT=0,
                active_area=pixel.active_area,
                save_plots=False)

pixel = pixels[px]
switch_pixel(pixel.connection)

illum = 'light'
folder = 'Dark_Voc_recovery'

for j in range(recovery_iterations):

    time.sleep(t_recovery[j])

    pico.turn_on()

    keithley.sweep_jv(output=Path(pixel.temppath),
                    experimentname=experimentname,
                    devicename=pixel.name.format(folder,illum,str(sum(t_recovery[:j+1])).zfill(5)),
                    led=led,
                    t_lightsoaking=sum(t_recovery[:j+1]),
                    led_current=100,
                    temperature='RT',
                    start_voltage=-0.2,
                    end_voltage=1.2,
                    voltage_step=0.04,
                    scan_speed=0.1,
                    direction='bidirectional',
                    prebiasV=0,
                    prebiasT=0,
                    active_area=pixel.active_area,
                    save_plots=False)

    pico.turn_off()

illum = 'dark'
t = '{}'.format(str(sum(t_recovery)).zfill(5))
folder = 'Dark_Voc_recovery'

```

```
for pixel in pixels:
    switch_pixel(pixel.connection)

    keithley.sweep_jv(output=Path(pixel.temppath),
                experimentname=experimentname,
                devicename=pixel.name.format(folder,illum,t),
                led=led,
                t_lightsoaking=0,
                led_current=100,
                temperature='RT',
                start_voltage=-0.2,
                end_voltage=1.2,
                voltage_step=0.04,
                scan_speed=0.1,
                direction='bidirectional',
                prebiasV=0,
                prebiasT=0,
                active_area=pixel.active_area,
                save_plots=False)

#%% Separate recovery (for after light soaking outside GB?)
illum = 'light'
folder = 'x'
for j in range(recovery_iterations):

    time.sleep(t_recovery[j])

    pico.turn_on()

    for pixel in pixels:
        switch_pixel(pixel.connection)

        keithley.sweep_jv(output=Path(pixel.temppath),
                        experimentname=experimentname,
                        devicename=pixel.name.format(folder,illum,sum(t_recovery[:j+1])),
                        led=led,
                        t_lightsoaking=sum(t_recovery[:j+1]),
                        led_current=100,
                        temperature='RT',
                        start_voltage=-0.2,
                        end_voltage=1.2,
                        voltage_step=0.04,
                        scan_speed=0.1,
                        direction='bidirectional',
                        prebiasV=0,
                        prebiasT=0,
                        active_area=pixel.active_area,
                        save_plots=False)

    pico.turn_off()

#%% End
```

E_plotter_API.py

```
import numpy as np
import matplotlib.pyplot as plt
import os
import pandas as pd
from scipy.interpolate import interp1d

n = 9 #Number of columns desired for results list, determined by number of parameters to be calculated + sample names (Voc, Jsc, FF,
     PCE, forward and reverse for all)
```

```
results = [[] for i in range(n)]
area = 0.0625 #cm2

class E_plotter:
    def __init__(self, file_path):
        self.data = []
        self.voltage = []
        self.current = []
        self.currdens = []
        self.time = []
        self.fpath = file_path
        self.results = [[] for i in range(n)]
        os.chdir(file_path)
        print('Files in the folder:')
        for i in sorted(os.listdir()):
            print(i)

    def load_data(self, filename):
        tab = pd.read_csv(filename, skiprows=20, nrows=1, sep='\t').shape[1]
        com = pd.read_csv(filename, skiprows=20, nrows=1, sep=',').shape[1]
        #space = pd.read_csv(filename, skiprows=20, nrows=1, sep='\s+').shape[1]
        #print(tab,com)

        if tab > com: #and tab > space:
            data = pd.read_csv(filename, skiprows=19, sep='\t')

        elif com > tab: #and com > space:
            data = pd.read_csv(filename, sep=',')

        else:
            print('Could not read file')
        # elif space > tab and space > com:
        # #print(pd.read_csv(filename, nrows=1, header=None, sep='\s+')[0].iloc[0])
        # if isinstance(pd.read_csv(filename, nrows=1, header=None, sep='\s+')[0].iloc[0], str):
        # data = pd.read_csv(filename, sep='\s+')
        # data.columns=['time', 'target voltage', 'current', 'voltage']
        # else:
        # data = pd.read_csv(filename, sep='\s+', header=None)
        # data.columns=['time', 'current', 'voltage']

        self.data = data

        if 'SMU1 Voltage' in data.columns:
            self.voltage = data['SMU1 Voltage']
            self.current = data['SMU1 Current']
            self.time = data['SMU1 Time']

        elif 'Sweep cyclic [V]' in data.columns:
            self.voltage = data['Sweep cyclic [V]']
            self.current = data['Current [A]']
            self.currdens = data['J [mA/cm]']

        elif 'voltage' in data.columns:
            self.voltage = data['voltage']
            self.current = data['current']
            self.time = data['time']

        elif 'Voltage (V)' in data.columns:
            self.voltage = data['Voltage (V)']
            self.current = data['Current (A)']
            self.currdens = data['Current (A)'] *10**3 / area

        else:
            print('Could not find parameters')

```

```
        #return data

    def plot_IV(self, voltage, current, name, multiple_Cycles=False, show_plots=False, save_plots=False):

        if multiple_Cycles == False:

            fig, ax = plt.subplots(num=name+'_lin_I-V')
            ax.plot(voltage, current, c='black')
            ax.set_ylabel('Current (A)')
            ax.set_xlabel('Voltage (V)')
            fig.tight_layout()
            if save_plots == True:
                plt.savefig(name+'_lin_I-V.png', dpi=150)
            if show_plots ==True:
                plt.show()
            plt.close()

            fig, ax = plt.subplots(num=name+'_log_I-V.png')
            ax.plot(voltage, abs(current), c='black')
            ax.set_yscale('log')
            ax.set_ylabel('Current (A)')
            ax.set_xlabel('Voltage (V)')
            fig.tight_layout()
            if save_plots == True:
                plt.savefig(name+'_log_I-V.png', dpi=150)

            if show_plots ==True:
                plt.show()
            plt.close()

        elif multiple_Cycles == True:

            colors = plt.get_cmap('hot')

            cycleStart = voltage[voltage.isin([0])][::2]

            fig, ax = plt.subplots(num=name+'_lin_I-V')

            for k in range(0,len(cycleStart)):
                if k < len(cycleStart.index)-1:
                    ax.plot(voltage.loc[cycleStart.index[k]:cycleStart.index[k+1]],
                         current.loc[cycleStart.index[k]:cycleStart.index[k+1]], label='Cycle '+ str(k+1),
                         c=colors((6/7)*k/len(cycleStart)))
                else:
                    ax.plot(voltage.loc[cycleStart.index[k]:], current.loc[cycleStart.index[k]:], label='Cycle '+ str(k+1),
                         c=colors((6/7)*k/len(cycleStart)))

            ax.set_xlabel('Voltage (V)')
            ax.set_ylabel('Current (A)')
            ax.legend()
            fig.tight_layout()
            plt.savefig(name+'_lin_I-V.png', dpi=150)
            if show_plots ==True:
                plt.show()
            plt.close()

            fig, ax = plt.subplots(num=name+'_log_I-V')
            for k in range(0,len(cycleStart)):
                if k < len(cycleStart.index)-1:
                    ax.plot(voltage.loc[cycleStart.index[k]:cycleStart.index[k+1]],
                         abs(current.loc[cycleStart.index[k]:cycleStart.index[k+1]]), label='Cycle '+ str(k+1),
                         c=colors((6/7)*k/len(cycleStart)))
                else:
                    ax.plot(voltage.loc[cycleStart.index[k]:], abs(current.loc[cycleStart.index[k]:]), label='Cycle '+ str(k+1),
                         c=colors((6/7)*k/len(cycleStart)))
```

```
            ax.set_yscale('log')

        ax.set_xlabel('Voltage (V)')
        ax.set_ylabel('Current (A)')
        ax.legend()
        fig.tight_layout()
        if save_plots == True:
            plt.savefig(name+'_log_I-V.png', dpi=150)
        if show_plots ==True:
            plt.show()
        plt.close()

def plot_JV(self, voltage, currDens, name, multiple_Cycles=False, save_results=True, show_plots=True, save_plots=False): #Nested list to collect results

    if multiple_Cycles == False:
        fig, ax = plt.subplots(num=name+'_lin_J-V')
        ax.plot(voltage, currDens, c='red')
        ax.plot(voltage, currDens, c='red')

        ax.set_xlabel('Voltage (V)')
        ax.set_ylabel('Current density (mA/cm$^2$)')
        fig.tight_layout()
        plt.axhline(y = 0, color = 'black', linestyle = '-')
        plt.title(name)

        forwV_index = int(len(voltage)/2 + 1) #final data point of forward scan V
        forwJ_index = int(len(currDens)/2 + 1) #final data point of forward scan J
        revV_index = int(len(voltage)) #final data point of reverse scan V
        revJ_index = int(len(currDens)) #final data point of reverse scan J

        # interpolation of both J and V, to get values between measured points
        interp_forwJ = interp1d(voltage[:forwV_index],currDens[:forwJ_index])
        interp_revJ = interp1d(voltage[forwV_index-1:revV_index],currDens[forwJ_index-1:revJ_index])
        # ax.plot(voltage[:forwV_index],interp_forwJ(voltage[:forwV_index]), c='blue') #Shows forward curve after interpolation of J
        # ax.plot(voltage[forwV_index-1:revV_index],interp_revJ(voltage[forwV_index-1:revV_index]), c='blue') #Shows reverse curve after interpolation of J

        interp_forwV = interp1d(currDens[:forwJ_index],voltage[:forwV_index]) #fill_value='extrapolate' can avoid ValueError, but use with caution
        interp_revV = interp1d(currDens[forwJ_index-1:revJ_index],voltage[forwV_index-1:revV_index])
        # ax.plot(interp_forwV(currDens[:forwJ_index]),currDens[:forwJ_index], c='green') #Shows forward curve after interpolation of V
        # ax.plot(interp_revV(currDens[forwJ_index-1:revJ_index]),currDens[forwJ_index-1:revJ_index],c='green') #Shows reverse curve after interpolation of V

        #calculation of Voc, Jsc, FF, MPP and PCE for light curves
        if 'light' in name: #To make sure Voc, Jsc, FF and PCE are only calculated for light curves
            if max(currDens)>0: #Skips pixels resulting in 0 current
                Voc_forw = interp_forwV(0) #Calculates forward Voc from interpolated values
                Voc_rev = interp_revV(0) #Calculates reverse Voc from interpolated values
                plt.plot(Voc_forw, 0, marker='x') #Plots marker at forward Voc
                plt.plot(Voc_rev, 0, marker='x') #Plots marker at reverse Voc

                Jsc_forw = interp_forwJ(0) #Calculates forward Jsc from interpolated values
                Jsc_rev = interp_revJ(0) #Calculates reverse Jsc from interpolated values
                plt.plot(0,Jsc_forw, marker='x') #Plots marker at forward Jsc
                plt.plot(0,Jsc_rev, marker='x') #Plots marker at reverse Jsc

                power_forw = interp_forwJ(voltage[:forwV_index])*interp_forwV(currDens[:forwJ_index]) #Calculates power from forward curve (J*V at every point)
                power_rev = interp_revJ(voltage[forwV_index-1:revV_index])*interp_revV(currDens[forwJ_index-1:revJ_index]) #Calculates power from reverse curve (J*V at every point)
                MPP_forw = min(power_forw) #Calculates forward MPP by finding minimum of calculated power curve
                MPP_rev = min(power_rev) #Calculates reverse MPP by finding minimum of calculated power curve
```

```
            loc_MPP_forw = np.where(power_forw == MPP_forw) #List of indeces of where calculated values for the power are equal
                to the forward MPP
            loc_MPP_rev = np.where(power_rev == MPP_rev) #List of indeces of where calculated values for the power are equal to
                the reverse MPP
            plt.plot(interp_forwV(currDens[:forwJ_index])[loc_MPP_forw[0][0]],interp_forwJ(voltage[:forwV_index])[loc_MPP_forw[0][0]],
                marker='x') #Plots marker at location of MPP, by taking the first index in loc_MPP_forw
            plt.plot(interp_revV(currDens[forwJ_index-1:revJ_index])[loc_MPP_rev[0][0]],interp_revJ(voltage[forwV_index-1:revV_index])[l
                marker='x') #Plots marker at location of MPP, by taking the first index in loc_MPP_rev

            FF_forw = (MPP_forw)/(Jsc_forw*Voc_forw) * 100 #Calculates forward FF from interpolated values
            FF_rev = (MPP_rev)/(Jsc_rev*Voc_rev) * 100 #Calculates reverse FF from interpolated values

            P_in = 100 # mW cm^-2 #Power sun
            PCE_forw = (abs(Jsc_forw)*Voc_forw*FF_forw) / (P_in) #Calculates forward PCE from interpolated values
            PCE_rev = (abs(Jsc_rev)*Voc_rev*FF_rev) / (P_in) #Calculates forward PCE from interpolated values

            #Prints calculated parameters in plots
            textstr = '\n'.join(('Voc forw: {}'.format(np.around(Voc_forw,decimals=2)), 'Voc rev:
                {}'.format(np.around(Voc_rev,decimals=2)), 'Jsc forw: {}'.format(np.around(Jsc_forw,decimals=2)), 'Jsc rev:
                {}'.format(np.around(Jsc_rev,decimals=2)), 'FF forw: {}'.format(np.around(FF_forw,decimals=2)), 'FF rev:
                {}'.format(np.around(FF_rev,decimals=2)), 'PCE forw: {}'.format(np.around(PCE_forw,decimals=2)), 'PCE rev:
                {}'.format(np.around(PCE_rev,decimals=2))))
            ax.text(min(voltage)-0.03, max(currDens), textstr, verticalalignment = 'top',
                bbox={'facecolor':'none','alpha':0.5,'pad':5})

            if save_results == True:
                #Appends calculated parameters to results list
                results[0].append(name)
                results[1].append(Voc_forw)
                results[2].append(Voc_rev)
                results[3].append(Jsc_forw)
                results[4].append(Jsc_rev)
                results[5].append(FF_forw)
                results[6].append(FF_rev)
                results[7].append(PCE_forw)
                results[8].append(PCE_rev)

        self.results = results

        if save_plots == True:
            plt.savefig(name+'_lin_J-V.png', dpi=150)
        if show_plots == True:
            plt.show()
        plt.close()

        # fig, ax = plt.subplots(num=name+'_log_J-V')
        # ax.plot(voltage, abs(currDens), c='black')
        # ax.set_yscale('log')
        # ax.plot(voltage, currDens, c='black')
        # ax.set_xlabel('Voltage (V)')
        # ax.set_ylabel('Current density (mA/cm$^2$)')
        # fig.tight_layout()
        # if save_plots == True:
        # plt.savefig(name+'_log_J-V.png', dpi=150)
        # if show_plots == True:
        # plt.show()
        # plt.close()

    elif multiple_Cycles == True:
        cycleStart = voltage[voltage.isin([0])][::2]
        colors = plt.get_cmap('hot')
        fig, ax = plt.subplots(num=name+'_lin_J-V')

        for k in range(0,len(cycleStart)):
```

```
            if k < len(cycleStart.index)-1:
                ax.plot(voltage.loc[cycleStart.index[k]:cycleStart.index[k+1]],
                    currDens.loc[cycleStart.index[k]:cycleStart.index[k+1]], label='Cycle '+ str(k+1),
                    c=colors((6/7)*k/len(cycleStart)))
            else:
                ax.plot(voltage.loc[cycleStart.index[k]:], currDens.loc[cycleStart.index[k]:], label='Cycle '+ str(k+1),
                    c=colors((6/7)*k/len(cycleStart)))

        ax.set_xlabel('Voltage (V)')
        ax.set_ylabel('Current density (mA/cm$^2$)')
        ax.legend()
        fig.tight_layout()
        if save_plots == True:
            plt.savefig(name+'_lin_J-V.png', dpi=150)
        if show_plots == True:
            plt.show()
        plt.close()

        fig, ax = plt.subplots(num=name+'_log_J-V')
        for k in range(0,len(cycleStart)):
            if k < len(cycleStart.index)-1:
                ax.plot(voltage.loc[cycleStart.index[k]:cycleStart.index[k+1]],
                    abs(currDens.loc[cycleStart.index[k]:cycleStart.index[k+1]]), label='Cycle '+ str(k+1),
                    c=colors((6/7)*k/len(cycleStart)))
            else:
                ax.plot(voltage.loc[cycleStart.index[k]:], abs(currDens.loc[cycleStart.index[k]:]), label='Cycle '+ str(k+1),
                    c=colors((6/7)*k/len(cycleStart)))
            ax.set_yscale('log')

        ax.set_xlabel('Voltage (V)')
        ax.set_ylabel('Current density (mA/cm$^2$)')
        ax.legend()
        fig.tight_layout()
        if save_plots == True:
            plt.savefig(name+'_log_J-V.png', dpi=150)
        if show_plots == True:
            plt.show()
        plt.close()

    def plot_pulses(self, voltage, current, time, name,show_plots=False, save_plots=False):
        fig, ax1 = plt.subplots()
        ax1.plot(time, voltage, c='firebrick')
        ax1.set_xlabel('Time(s)')
        ax1.set_ylabel('Voltage (V)', color='firebrick')
        ax1.tick_params(axis='y', labelcolor='firebrick')

        ax2 = ax1.twinx()
        ax2.plot(time, current, c='royalblue')
        ax2.set_ylabel('Current (A)', color='royalblue')
        ax2.tick_params(axis='y', labelcolor='royalblue')

        #Align the 0 levels of both y-axes
        # m1, M1 = ax1.get_ylim() ; ticks1 = ax1.get_yticks()
        # m2, M2 = ax2.get_ylim() ; ticks2 = ax2.get_yticks()

        # w1 = abs(m1)/(M1+abs(m1))
        # w2 = abs(m2)/(M2+abs(m2))

        # M1 = (m1*M2/m2 if w2<=w1 else M1 )
        # m1 = (m1 if w2<=w1 else M1*m2/M2)

        # ax1.set_ylim(m1,M1) ; ax1.set_yticks(ticks1)
        # ax2.set_ylim(m2,M2) ; ax2.set_yticks(ticks2)

        fig.tight_layout()
```

```
        if save_plots == True:
            plt.savefig(name +'_I-V-t.png', dpi=150)
        if show_plots == True:
                plt.show()
        plt.close()
```

lightsoaking_plotting.py

```
    # -*- coding: utf-8 -*-
"""
Created on Tue May 30 17:31:05 2023

@author: ddekker
"""

import numpy as np
import matplotlib.pyplot as plt
import matplotlib.colors as col
import os
from E_plotter_API import E_plotter

auto = E_plotter(os.path.dirname(__file__))

order = [2,4,9,1,3,10,20,26,7,22,28,32] #[1,5,2,7,4] #change for every batch!!! Can put in the right order

#%% Prep for plots
labels = []
time = []
t_lightsoaking = [0] + [600]*12 + [3600]*10 #list with all the light soaking intervals between JV measurements
for i in np.arange(0,len(t_lightsoaking),1):
    x = np.round(sum(t_lightsoaking[:i+1])/3600,2) #calculates time point of each JV measurement in hours
    labels.append('{} h'.format(x)) #list of strings with all the time points of the individual JV measurements (in hours) during light soaking
    time.append(x) #list of floats with all the time points of the individual JV measurements (in hours) during light soaking

labels_rec = []
time_rec = []
t_recovery = [300]*4 + [600]*4 + [3600]*11 #list with all the recovery intervals between JV measurements
for i in np.arange(0,len(t_recovery),1):
    x = np.round(sum(t_recovery[:i+1])/3600,2) #calculates time point of each JV measurement in hours excluding light soaking time
    labels_rec.append('{} h'.format(x)) #list of strings with all the time points of the individual JV measurements (in hours) during recovery

    y = time[-1:] + np.round(sum(t_recovery[:i+1])/3600,2) #calculates time point of each JV measurement including light soaking time
    time_rec.append(y) #list of floats with all the time points of the individual JV measurements (in hours) during recovery

redD = col.ListedColormap(('#f2daff','#ead0f8','#e3c6f2','#dbbceb','#d3b2e5','#cba8de','#c49fd8','#bc95d1','#b48bcb','#ac81c4','#a577be','#9d6db

#redD = col.ListedColormap(('#e8ecfb', '#d0d6f7', '#b9c0f3', '#a1aaef', '#8994eb', '#718ee7', '#5a78e3', '#4262df', '#2a4cdb',
     '#1246d7', '#003fd0', '#0039bb', '#0033a7', '#002d92', '#00277e', '#002169', '#001b55', '#001540', '#000f2c', '#000917',
     '#000302'))
redD_rec = col.ListedColormap(('#f2daff','#e9cef7','#dfc2ef','#d6b6e7','#ccaadf','#c39dd7','#b991cf','#b085c7','#a679bf','#9d6db7','#9461af','#8a55a

greenD = col.ListedColormap(('#e9feff',
'#daf9fb',
'#ccf4f6',
'#bdeff2',
'#afeaed',
'#a0e5e9',
'#92e0e4',
'#83dbe0',
```

```
'#75d6db',
'#66d1d7',
'#58ccd2',
'#49c7ce',
'#43bdc5',
'#3db2bb',
'#37a8b2',
'#319da8',
'#2b939f',
'#248895',
'#1e7e8c',
'#187382',
'#126979',
'#0c5e6f',
'#065466'))

greenD_rec = col.ListedColormap(('#e9feff',
'#d7f8fa',
'#c5f2f4',
'#b4ecef',
'#a2e6e9',
'#90dfe4',
'#7ed9de',
'#6dd3d9',
'#5bcdd3',
'#49c7ce',
'#42bac2',
'#3aadb7',
'#33a1ab',
'#2b94a0',
'#248794',
'#1c7a89',
'#156e7d',
'#0d6172',
'#065466'))

pinkD = col.ListedColormap(('#ffe9f1',
'#fbd6e3',
'#f6c2d5',
'#f2afc7',
'#ed9bb9',
'#e988ab',
'#e4749e',
'#e06190',
'#db4d82',
'#d73a74',
'#d22666',
'#ce1358',
'#c51253',
'#bb114d',
'#b21048',
'#a80f43',
'#9f0e3e',
'#950c38',
'#8c0b33',
'#820a2e',
'#790929',
'#6f0823',
'#66071e'))

pinkD_rec = col.ListedColormap(('#ffe9f1',
'#fad1e0',
'#f4b9cf',
'#efa2be',
'#e98aad',
```

```
'#e4729c',
'#de5a8b',
'#d9437a',
'#d32b69',
'#ce1358',
'#c21252',
'#b7104b',
'#ab0f45',
'#a00e3e',
'#940c38',
'#890b31',
'#7d0a2b',
'#720824',
'#66071e'))

#%% Make colorbar (incorrect labels)

arr = plt.imshow([time],cmap=greenD)
fig10 = plt.figure(figsize=(6,4), dpi=300)
cb = plt.colorbar(arr, label='Time (h)')
cb.minorticks_on()

plt.show()

#%% Functions to plot JV curves over time
def plot_curves(composition,condition,pixel):

    fig1 = plt.figure(figsize=(6,4), dpi=300)
    ax1 = plt.gca()

    fontsize1 = 13
    ax1.set_xlim(-0.25,0.9)
    ax1.set_ylim(-35,40)
    ax1.tick_params(length=5,labelsize=fontsize1)

    if composition == 'EDAI':
        colors = pinkD(np.linspace(0, 1, len(labels)))
        if condition == 'Light_Voc':
            device = order[2] #change for every batch
        elif condition == 'Light_0V':
            device = order[5] #change for every batch
        elif condition == 'Dark_0V':
            device = order[8] #change for every batch
        elif condition == 'Dark_Voc':
            device = order[11] #change for every batch

    elif composition == 'Unpassivated':
        colors = greenD(np.linspace(0, 1, len(labels)))
        if condition == 'Light_Voc':
            device = order[1] #change for every batch
        elif condition == 'Light_0V':
            device = order[4] #change for every batch
        elif condition == 'Dark_0V':
            device = order[7] #change for every batch
        elif condition == 'Dark_Voc':
            device = order[10] #change for every batch

    elif composition == 'Salts':
        colors = redD(np.linspace(0, 1, len(labels)))
        if condition == 'Light_Voc':
            device = order[0] #change for every batch
        elif condition == 'Light_0V':
            device = order[3] #change for every batch
        elif condition == 'Dark_0V':
            device = order[6] #change for every batch
```

```
        elif condition == 'Dark_Voc':
            device = order[9] #change for every batch

    j = 0
    for i in sorted(os.listdir()):
        if i.startswith('{}_d{}_light'.format(condition,str(device).zfill(2))):
            if i.endswith('0_P{}.csv'.format(pixel)):
                auto.load_data(i)
                print('Plotting data from', i)

                if 'Voltage (V)' in auto.data.columns:
                    ax1.plot(auto.voltage,auto.currdens,color=colors[j],label=labels[j])
                    j = j + 1

    plt.axhline(y = 0, color = 'black', linestyle = '-',linewidth=0.1)

    ax1.set_xlabel('Voltage (V)',fontsize=fontsize1)
    ax1.set_ylabel('Current density (mA/cm$^2$)',fontsize=fontsize1)
    plt.title('{} {} pixel {}'.format(condition,composition,pixel))

    plt.show()

def plot_curves_rec(composition,condition,pixel):

    fig3 = plt.figure(figsize=(6,4), dpi=300)
    ax3 = plt.gca()

    fontsize1 = 13
    ax3.set_xlim(-0.25,0.9)
    ax3.set_ylim(-35,40)
    ax3.tick_params(length=5,labelsize=fontsize1)


    if composition == 'EDAI':
        colors = pinkD_rec(np.linspace(0, 1, len(labels_rec)))
        if condition == 'Light_Voc':
            device = order[2] #change for every batch
        elif condition == 'Light_0V':
            device = order[5] #change for every batch
        elif condition == 'Dark_0V':
            device = order[8] #change for every batch
        elif condition == 'Dark_Voc':
            device = order[11] #change for every batch

    elif composition == 'Unpassivated':
        colors = greenD_rec(np.linspace(0, 1, len(labels_rec)))
        if condition == 'Light_Voc':
            device = order[1] #change for every batch
        elif condition == 'Light_0V':
            device = order[4] #change for every batch
        elif condition == 'Dark_0V':
            device = order[7] #change for every batch
        elif condition == 'Dark_Voc':
            device = order[10] #change for every batch

    elif composition == 'Salts':
        colors = redD_rec(np.linspace(0, 1, len(labels_rec)))
        if condition == 'Light_Voc':
            device = order[0] #change for every batch
        elif condition == 'Light_0V':
            device = order[3] #change for every batch
        elif condition == 'Dark_0V':
            device = order[6] #change for every batch
        elif condition == 'Dark_Voc':
            device = order[9] #change for every batch
```

```
245
246     j = 0
247     for i in sorted(os.listdir()):
248         if i.startswith('{}_recovery_d{}_light'.format(condition,str(device).zfill(2))):
249             if i.endswith('0_P{}.csv'.format(pixel)):
250                 auto.load_data(i)
251                 print('Plotting data from', i)
252
253                 if 'Voltage (V)' in auto.data.columns:
254                     ax3.plot(auto.voltage,auto.currdens,color=colors[j],label=labels_rec[j])
255                     j = j + 1
256
257     plt.axhline(y = 0, color = 'black', linestyle = '-',linewidth=0.1)
258
259     ax3.set_xlabel('Voltage (V)',fontsize=fontsize1)
260     ax3.set_ylabel('Current density (mA/cm$^2$)',fontsize=fontsize1)
261     plt.title('Recovery {} {} pixel {}'.format(condition,composition,pixel))
262     plt.show()
263
264 #%% Plotting JV curves of one pixel over time
265 comp = 'EDAI'
266 cond = 'Dark_Voc'
267 px = 4
268
269 plot_curves(composition=comp,condition=cond,pixel=px)
270 plot_curves_rec(composition=comp,condition=cond,pixel=px)
271
272 #%% Plot individual JV curves and calculate Voc, Jsc, FF, PCE
273 for x in order:
274     if x == order[0] or x == order[1] or x == order [2]:
275         y = 'Light_Voc'
276     elif x == order[3] or x == order[4] or x == order[5]:
277         y = 'Light_0V'
278     elif x == order[6] or x == order[7] or x == order[8]:
279         y = 'Dark_0V'
280     elif x == order[9] or x == order[10] or x == order[11]:
281         y = 'Dark_Voc'
282
283     for i in sorted(os.listdir()): #careful, if you run this twice, it will not overwrite the results but append them to the same list,
          useful for appending multiple devices to the list, if you want to wipe it, use ctrl+W
284         if '_2_P' in i:
285             continue
286         elif i.startswith('{}_d{}_light_'.format(y,str(x).zfill(2))):
287             auto.load_data(i)
288             print('Plotting data from', i)
289             auto.plot_JV(voltage=auto.voltage,currDens=auto.currdens,name=i,save_results=True) #use save_results=False to just plot
                  without appending results to list
290
291 #%% Averaging of all parameters
292 devices = [''] #List will be header containing device number in results file
293
294 Voc_forw_final = [] #Empty lists to collect a specific parameter of every pixel of every device after removal of outliers
295 Voc_rev_final = []
296 Jsc_forw_final = []
297 Jsc_rev_final = []
298 FF_forw_final = []
299 FF_rev_final = []
300 PCE_forw_final = []
301 PCE_rev_final = []
302
303 Voc_forw_final_avg = ['Voc forward'] #Lists to collect averages of a specific parameter of all pixels in one device for all devices
304 Voc_rev_final_avg = ['Voc reverse']
305 Jsc_forw_final_avg = ['Jsc forward']
306 Jsc_rev_final_avg = ['Jsc reverse']
307 FF_forw_final_avg = ['FF forward']
```

```
FF_rev_final_avg = ['FF reverse']
PCE_forw_final_avg = ['PCE forward']
PCE_rev_final_avg = ['PCE reverse']
Voc_forw_std = ['Standard deviation forward Voc'] #Lists to collect standard deviation in all parameters
Voc_rev_std = ['Standard deviation reverse Voc']
Jsc_forw_std = ['Standard deviation forward Jsc']
Jsc_rev_std = ['Standard deviation reverse Jsc']
FF_forw_std = ['Standard deviation forward FF']
FF_rev_std = ['Standard deviation reverse FF']
PCE_forw_std = ['Standard deviation forward PCE']
PCE_rev_std = ['Standard deviation reverse PCE']
nr_used_px = ['Number of pixels used']


for i in order:#np.arange(1,nr_of_devices+1,1): #Loops over all devices to average parameters of all pixels in one device
    name_nr = str(i).zfill(2)
    startd4 = [auto.results[0].index(x) for x in auto.results[0] if 'd{}'.format(name_nr) in x][0] #Finds first pixel of one device
    endd4 = [auto.results[0].index(x) for x in auto.results[0] if 'd{}'.format(name_nr) in x][-1] + 1 #Finds last pixel of that device
    print('i',startd4,endd4)

    for l in np.arange(0,len(t_lightsoaking),1): #separates all subsequent measurements of same device during light soaking
        name_nr_2 = str(sum(t_lightsoaking[:l+1])).zfill(5)
        start4 = [auto.results[0].index(x) for x in auto.results[0][startd4:endd4] if '{}'.format(name_nr_2) in x][0]
        end4 = [auto.results[0].index(x) for x in auto.results[0][startd4:endd4] if '{}'.format(name_nr_2) in x][-1] + 1
        print('l',start4,end4)
        devices.append('d{} t{}'.format(i,l)) #Appends device number to header list

        Voc_forw_final = [x for x in auto.results[1][start4:end4]] #Selects the forward Voc of all pixels in one device
        Voc_rev_final = [x for x in auto.results[2][start4:end4]] #Selects the reverse Voc of all pixels in one device
        Jsc_forw_final = [x for x in auto.results[3][start4:end4]] #Selects the forward Jsc of all pixels in one device
        Jsc_rev_final = [x for x in auto.results[4][start4:end4]] #Selects the reverse Jsc of all pixels in one device
        FF_forw_final = [x for x in auto.results[5][start4:end4]] #Selects the forward FF of all pixels in one device
        FF_rev_final = [x for x in auto.results[6][start4:end4]] #Selects the reverse FF of all pixels in one device
        PCE_forw_final = [x for x in auto.results[7][start4:end4]] #Selects the forward PCE of all pixels in one device
        PCE_rev_final = [x for x in auto.results[8][start4:end4]] #Selects the reverse PCE of all pixels in one device

        if i == 7:
            for k in [1,0]:
                #if FF_forw_final[k] < 35 or FF_rev_final[k] < 35:
                Voc_forw_final = np.delete(Voc_forw_final,k)
                Voc_rev_final = np.delete(Voc_rev_final,k)
                Jsc_forw_final = np.delete(Jsc_forw_final,k)
                Jsc_rev_final = np.delete(Jsc_rev_final,k)
                FF_forw_final = np.delete(FF_forw_final,k)
                FF_rev_final = np.delete(FF_rev_final,k)
                PCE_forw_final = np.delete(PCE_forw_final,k)
                PCE_rev_final = np.delete(PCE_rev_final,k)

        # if i == 6:
        # for k in [2,1,0]:
        # #if FF_forw_final[k] < 35 or FF_rev_final[k] < 35:
        # Voc_forw_final = np.delete(Voc_forw_final,k)
        # Voc_rev_final = np.delete(Voc_rev_final,k)
        # Jsc_forw_final = np.delete(Jsc_forw_final,k)
        # Jsc_rev_final = np.delete(Jsc_rev_final,k)
        # FF_forw_final = np.delete(FF_forw_final,k)
        # FF_rev_final = np.delete(FF_rev_final,k)
        # PCE_forw_final = np.delete(PCE_forw_final,k)
        # PCE_rev_final = np.delete(PCE_rev_final,k)

        Voc_forw_final_avg.append(np.round(sum(Voc_forw_final)/len(Voc_forw_final),2)) #Calculates the average forward Voc of one device
            from the corrected data set
        Voc_rev_final_avg.append(np.round(sum(Voc_rev_final)/len(Voc_rev_final),2)) #Calculates the average reverse Voc of one device
            from the corrected data set
        Jsc_forw_final_avg.append(np.round(sum(Jsc_forw_final)/len(Jsc_forw_final),2)) #Calculates the average forward Jsc of one device
```

```
            from the corrected data set
        Jsc_rev_final_avg.append(np.round(sum(Jsc_rev_final)/len(Jsc_rev_final),2)) #Calculates the average reverse Jsc of one device
            from the corrected data set
        FF_forw_final_avg.append(np.round(sum(FF_forw_final)/len(FF_forw_final),2)) #Calculates the average forward FF of one device
            from the corrected data set
        FF_rev_final_avg.append(np.round(sum(FF_rev_final)/len(FF_rev_final),2)) #Calculates the average reverse FF of one device from
            the corrected data set
        PCE_forw_final_avg.append(np.round(sum(PCE_forw_final)/len(PCE_forw_final),2)) #Calculates the average forward PCE of one device
            from the corrected data set
        PCE_rev_final_avg.append(np.round(sum(PCE_rev_final)/len(PCE_rev_final),2)) #Calculates the average reverse PCE of one device
            from the corrected data set

        Voc_forw_std.append(np.round(np.std(Voc_forw_final),2))
        Voc_rev_std.append(np.round(np.std(Voc_rev_final),2))
        Jsc_forw_std.append(np.round(np.std(Jsc_forw_final),2))
        Jsc_rev_std.append(np.round(np.std(Jsc_rev_final),2))
        FF_forw_std.append(np.round(np.std(FF_forw_final),2))
        FF_rev_std.append(np.round(np.std(FF_rev_final),2))
        PCE_forw_std.append(np.round(np.std(PCE_forw_final),2)) #Calculates the standard deviation in the average forward PCE from the
            corrected data set
        PCE_rev_std.append(np.round(np.std(PCE_rev_final),2))
        nr_used_px.append(len(PCE_forw_final)) #Includes the number of pixels of each device that was used for the calculated averages

#Collects all averaged results in a numpy array, labeled with device number and name of parameter
results_avg =
     np.array([devices,Voc_forw_final_avg,Voc_rev_final_avg,Jsc_forw_final_avg,Jsc_rev_final_avg,FF_forw_final_avg,FF_rev_final_avg,PCE_forw_f

#%% RECOVERY: Plot individual JV curves and calculate Voc, Jsc, FF, PCE
for x in order:
    if x == order[0] or x == order[1] or x == order [2]:
        y = 'Light_Voc'
    elif x == order[3] or x == order[4] or x == order[5]:
        y = 'Light_OV'
    elif x == order[6] or x == order[7] or x == order[8]:
        y = 'Dark_OV'
    elif x == order[9] or x == order[10] or x == order[11]:
        y = 'Dark_Voc'

    for i in sorted(os.listdir()): #careful, if you run this twice, it will not overwrite the results but append them to the same list,
         useful for appending multiple devices to the list, if you want to wipe it, use ctrl+W
        if i.startswith('{}_recovery_d{}_light_'.format(y,str(x).zfill(2))):
            auto.load_data(i)
            print('Plotting data from', i)
            auto.plot_JV(voltage=auto.voltage,currDens=auto.currdens,name=i,save_results=True) #use save_results=False to just plot
                without appending results to list


#%% RECOVERY: Averaging of all parameters
devices_rec = [''] #List will be header containing device number in results file

Voc_forw_final_rec = [] #Empty lists to collect a specific parameter of every pixel of every device after removal of outliers
Voc_rev_final_rec = []
Jsc_forw_final_rec = []
Jsc_rev_final_rec = []
FF_forw_final_rec = []
FF_rev_final_rec = []
PCE_forw_final_rec = []
PCE_rev_final_rec = []

Voc_forw_final_avg_rec = ['Voc forward'] #Lists to collect averages of a specific parameter of all pixels in one device for all devices
Voc_rev_final_avg_rec = ['Voc reverse']
Jsc_forw_final_avg_rec = ['Jsc forward']
Jsc_rev_final_avg_rec = ['Jsc reverse']
FF_forw_final_avg_rec = ['FF forward']
FF_rev_final_avg_rec = ['FF reverse']
```

```
PCE_forw_final_avg_rec = ['PCE forward']
PCE_rev_final_avg_rec = ['PCE reverse']
Voc_forw_std_rec = ['Standard deviation forward Voc'] #Lists to collect standard deviation in all parameters
Voc_rev_std_rec = ['Standard deviation reverse Voc']
Jsc_forw_std_rec = ['Standard deviation forward Jsc']
Jsc_rev_std_rec = ['Standard deviation reverse Jsc']
FF_forw_std_rec = ['Standard deviation forward FF']
FF_rev_std_rec = ['Standard deviation reverse FF']
PCE_forw_std_rec = ['Standard deviation forward PCE']
PCE_rev_std_rec = ['Standard deviation reverse PCE']
nr_used_px_rec = ['Number of pixels used']


for i in order:#np.arange(1,nr_of_devices+1,1): #Loops over all devices to average parameters of all pixels in one device
    name_nr = str(i).zfill(2)
    startd4 = [auto.results[0].index(x) for x in auto.results[0] if 'recovery_d{}'.format(name_nr) in x][0] #Finds first pixel of one device
    endd4 = [auto.results[0].index(x) for x in auto.results[0] if 'recovery_d{}'.format(name_nr) in x][-1] + 1 #Finds last pixel of that device
    print('i',startd4,endd4)

    for l in np.arange(0,len(t_recovery),1): #separates all subsequent measurements of same device during light soaking
        name_nr_2 = str(sum(t_recovery[:l+1])).zfill(5)
        start4 = [auto.results[0].index(x) for x in auto.results[0][startd4:endd4] if '{}'.format(name_nr_2) in x][0]
        end4 = [auto.results[0].index(x) for x in auto.results[0][startd4:endd4] if '{}'.format(name_nr_2) in x][-1] + 1
        print('l',start4,end4)
        devices_rec.append('d{} t{}'.format(i,l)) #Appends device number to header list

        Voc_forw_final_rec = [x for x in auto.results[1][start4:end4]] #Selects the forward Voc of all pixels in one device
        Voc_rev_final_rec = [x for x in auto.results[2][start4:end4]] #Selects the reverse Voc of all pixels in one device
        Jsc_forw_final_rec = [x for x in auto.results[3][start4:end4]] #Selects the forward Jsc of all pixels in one device
        Jsc_rev_final_rec = [x for x in auto.results[4][start4:end4]] #Selects the reverse Jsc of all pixels in one device
        FF_forw_final_rec = [x for x in auto.results[5][start4:end4]] #Selects the forward FF of all pixels in one device
        FF_rev_final_rec = [x for x in auto.results[6][start4:end4]] #Selects the reverse FF of all pixels in one device
        PCE_forw_final_rec = [x for x in auto.results[7][start4:end4]] #Selects the forward PCE of all pixels in one device
        PCE_rev_final_rec = [x for x in auto.results[8][start4:end4]] #Selects the reverse PCE of all pixels in one device

        # if i == 5:
        # for k in [3]:
        # #if FF_forw_final[k]_rec < 35 or FF_rev_final[k]_rec < 35:
        # Voc_forw_final_rec = np.delete(Voc_forw_final_rec,k)
        # Voc_rev_final_rec = np.delete(Voc_rev_final_rec,k)
        # Jsc_forw_final_rec = np.delete(Jsc_forw_final_rec,k)
        # Jsc_rev_final_rec = np.delete(Jsc_rev_final_rec,k)
        # FF_forw_final_rec = np.delete(FF_forw_final_rec,k)
        # FF_rev_final_rec = np.delete(FF_rev_final_rec,k)
        # PCE_forw_final_rec = np.delete(PCE_forw_final_rec,k)
        # PCE_rev_final_rec = np.delete(PCE_rev_final_rec,k)
        if i == 7:
            for k in [1,0]:
                #if FF_forw_final[k]_rec < 35 or FF_rev_final[k]_rec < 35:
                Voc_forw_final_rec = np.delete(Voc_forw_final_rec,k)
                Voc_rev_final_rec = np.delete(Voc_rev_final_rec,k)
                Jsc_forw_final_rec = np.delete(Jsc_forw_final_rec,k)
                Jsc_rev_final_rec = np.delete(Jsc_rev_final_rec,k)
                FF_forw_final_rec = np.delete(FF_forw_final_rec,k)
                FF_rev_final_rec = np.delete(FF_rev_final_rec,k)
                PCE_forw_final_rec = np.delete(PCE_forw_final_rec,k)
                PCE_rev_final_rec = np.delete(PCE_rev_final_rec,k)

        Voc_forw_final_avg_rec.append(np.round(sum(Voc_forw_final_rec)/len(Voc_forw_final_rec),2)) #Calculates the average forward Voc of one device from the corrected data set
        Voc_rev_final_avg_rec.append(np.round(sum(Voc_rev_final_rec)/len(Voc_rev_final_rec),2)) #Calculates the average reverse Voc of one device from the corrected data set
        Jsc_forw_final_avg_rec.append(np.round(sum(Jsc_forw_final_rec)/len(Jsc_forw_final_rec),2)) #Calculates the average forward Jsc
```

```
            of one device from the corrected data set
        Jsc_rev_final_avg_rec.append(np.round(sum(Jsc_rev_final_rec)/len(Jsc_rev_final_rec),2)) #Calculates the average reverse Jsc of
            one device from the corrected data set
        FF_forw_final_avg_rec.append(np.round(sum(FF_forw_final_rec)/len(FF_forw_final_rec),2)) #Calculates the average forward FF of
            one device from the corrected data set
        FF_rev_final_avg_rec.append(np.round(sum(FF_rev_final_rec)/len(FF_rev_final_rec),2)) #Calculates the average reverse FF of one
            device from the corrected data set
        PCE_forw_final_avg_rec.append(np.round(sum(PCE_forw_final_rec)/len(PCE_forw_final_rec),2)) #Calculates the average forward PCE
            of one device from the corrected data set
        PCE_rev_final_avg_rec.append(np.round(sum(PCE_rev_final_rec)/len(PCE_rev_final_rec),2)) #Calculates the average reverse PCE of
            one device from the corrected data set

        Voc_forw_std_rec.append(np.round(np.std(Voc_forw_final_rec),2))
        Voc_rev_std_rec.append(np.round(np.std(Voc_rev_final_rec),2))
        Jsc_forw_std_rec.append(np.round(np.std(Jsc_forw_final_rec),2))
        Jsc_rev_std_rec.append(np.round(np.std(Jsc_rev_final_rec),2))
        FF_forw_std_rec.append(np.round(np.std(FF_forw_final_rec),2))
        FF_rev_std_rec.append(np.round(np.std(FF_rev_final_rec),2))
        PCE_forw_std_rec.append(np.round(np.std(PCE_forw_final_rec),2)) #Calculates the standard deviation in the average forward PCE
            from the corrected data set
        PCE_rev_std_rec.append(np.round(np.std(PCE_rev_final_rec),2))
        nr_used_px_rec.append(len(PCE_forw_final_rec)) #Includes the number of pixels of each device that was used for the calculated
            averages

#Collects all averaged results in a numpy array, labeled with device number and name of parameter
results_avg_rec =
     np.array([devices_rec,Voc_forw_final_avg_rec,Voc_rev_final_avg_rec,Jsc_forw_final_avg_rec,Jsc_rev_final_avg_rec,FF_forw_final_avg_rec,FF_

#%% Definition plotting functions
def plot_evolution(param, j_start):
    fig2 = plt.figure(figsize=(6,4),dpi=300)
    ax2 = plt.gca()

    fontsize2 = 13
    fontsize2 = 13
    elinewidth = 2
    capthick = 2
    capsize = 4
    linestyle = '-'
    marker = ['o','d']

    if param == 'Voc':
        y_forw = Voc_forw_final_avg
        y_rev = Voc_rev_final_avg
        yerr_forw = Voc_forw_std
        yerr_rev = Voc_rev_std
        unit = ' (V)'
        ylim=[0.45,0.85]

    elif param == 'Jsc':
        y_forw = Jsc_forw_final_avg
        y_rev = Jsc_rev_final_avg
        yerr_forw = Jsc_forw_std
        yerr_rev = Jsc_rev_std
        unit = ' (mA/cm$^2$)'
        ylim=[-11,-26]

    elif param == 'FF':
        y_forw = FF_forw_final_avg
        y_rev = FF_rev_final_avg
        yerr_forw = FF_forw_std
        yerr_rev = FF_rev_std
        unit = ' (%)'
        ylim=[35,75]
```

```
    elif param == 'PCE':
        y_forw = PCE_forw_final_avg
        y_rev = PCE_rev_final_avg
        yerr_forw = PCE_forw_std
        yerr_rev = PCE_rev_std
        unit = ' (%)'
        ylim = [4,13]

    ax2.set_ylim(ylim)
    #ax2.set_xlim([0,1])

    ax2.minorticks_on()
    ax2.tick_params(which='major',length=5,labelsize=fontsize2)
    ax2.tick_params(which='minor',length=3,labelsize=fontsize2)

    labels_2f = ['Salts forw','Unpassivated forw']
    colours_2f = ['thistle','turquoise']

    labels_2r = ['Salts rev','Unpassivated rev']
    colours_2r = ['rebeccapurple','teal']

    titles = ['Evolution of {} under continuous illumination at Voc','Evolution of {} under continuous illumination at 0V','Evolution
         of {} in the dark at 0V','Evolution of {} in the dark at 0.7V']
    if j_start == 1:
        title = titles[0]

    elif j_start == 70:
        title = titles[1]

    elif j_start == 139:
        title = titles[2]

    elif j_start == 208:
        title = titles[3]

        labels_2f = ['2PACz forw'] #because Dark 0V only measured for 2PACz
        colours_2f = ['turquoise']

        labels_2r = ['2PACz rev']
        colours_2r = ['teal']

        marker = ['d']

    j = j_start

    for i in np.arange(0,len(labels_2f),1):
        ax2.errorbar(time,y_forw[j:j+len(t_lightsoaking)],yerr=yerr_forw[j:j+len(t_lightsoaking)],elinewidth=elinewidth,capthick=capthick,capsi
        ax2.errorbar(time,y_rev[j:j+len(t_lightsoaking)],yerr=yerr_rev[j:j+len(t_lightsoaking)],elinewidth=elinewidth,capthick=capthick,capsize
        j = j + len(t_lightsoaking)

    ax2.set_xlabel('Time (h)',fontsize=fontsize2)
    ax2.set_ylabel(param+unit,fontsize=fontsize2)
    ax2.legend(loc='best',fontsize=fontsize2-2.5, ncols = 2)
    plt.title(title.format(param),fontsize=fontsize2)
    fig2.tight_layout()
    plt.show()

def plot_recovery(param, k_start):
    fig2 = plt.figure(figsize=(6,4),dpi=300)

    ax2 = plt.gca()
    fontsize2 = 13
    fontsize2 = 13
    elinewidth = 3
    capthick = 2
```

```
    capsize = 4
    linestyle = '-'
    marker = ['o','d']

    if param == 'Voc':
        y_forw = Voc_forw_final_avg_rec
        y_rev = Voc_rev_final_avg_rec
        yerr_forw = Voc_forw_std_rec
        yerr_rev = Voc_rev_std_rec
        unit = ' (V)'
        ylim=[0.45,0.85]

    elif param == 'Jsc':
        y_forw = Jsc_forw_final_avg_rec
        y_rev = Jsc_rev_final_avg_rec
        yerr_forw = Jsc_forw_std_rec
        yerr_rev = Jsc_rev_std_rec
        unit = ' (mA/cm$^2$)'
        ylim=[-11,-26]

    elif param == 'FF':
        y_forw = FF_forw_final_avg_rec
        y_rev = FF_rev_final_avg_rec
        yerr_forw = FF_forw_std_rec
        yerr_rev = FF_rev_std_rec
        unit = ' (%)'
        ylim=[35,75]

    elif param == 'PCE':
        y_forw = PCE_forw_final_avg_rec
        y_rev = PCE_rev_final_avg_rec
        yerr_forw = PCE_forw_std_rec
        yerr_rev = PCE_rev_std_rec
        unit = ' (%)'
        ylim = [1,14]

    ax2.set_ylim(ylim)
    ax2.set_xlim([11.5,24.5])

    ax2.minorticks_on()
    ax2.tick_params(which='major',length=5,labelsize=fontsize2)
    ax2.tick_params(which='minor',length=3,labelsize=fontsize2)

    labels_2f = ['IM/CF$_3$BOM forw','Unpassivated forw', 'EDAI forw']
    colours_2f = ['thistle','turquoise', 'pink']

    labels_2r = ['Salts rev','Unpassivated rev', 'EDAI rev']
    colours_2r = ['rebeccapurple','teal', 'mediumvioletred']

    titles = ['Recovery of {} after continuous illumination at Voc','Recovery of {} after continuous illumination at 0V','Recovery of
         {} after ageing in the dark at 0V','Recovery of {} after ageing in the dark at 0.7V']
    if k_start == 1:
        title = titles[0]

    elif k_start == 58:
        title = titles[1]

    elif k_start == 115:
        title = titles[2]

    elif k_start == 172:
        title = titles[3]

        labels_2f = ['2PACz forw'] #because Dark 0V only measured for 2PACz
        colours_2f = ['turquoise']
```

```
        labels_2r = ['2PACz rev']
        colours_2r = ['teal']

        marker = ['d']

    k = k_start

    for i in np.arange(0,len(labels_2f),1):
        ax2.errorbar(time_rec,y_forw[k:k+len(t_recovery)],yerr=yerr_forw[k:k+len(t_recovery)],elinewidth=elinewidth,capthick=capthick,capsize=c
        ax2.errorbar(time_rec,y_rev[k:k+len(t_recovery)],yerr=yerr_rev[k:k+len(t_recovery)],elinewidth=elinewidth,capthick=capthick,capsize=cap
        k = k + len(t_recovery)

    ax2.fill_between((-5,30),-30,80,alpha=0.35,color='#a8c8ec')
    ax2.set_xlabel('Time (h)',fontsize=fontsize2)
    ax2.set_ylabel(param+unit,fontsize=fontsize2)
    ax2.legend(loc='best',fontsize=fontsize2-2.5, ncols = 3)
    plt.title(title.format(param),fontsize=fontsize2)
    fig2.tight_layout()
    plt.show()

def plot_both(param, j_start, k_start):
    fig2 = plt.figure(figsize=(14,8),dpi=600)
    ax2 = plt.gca()

    fontsize2 = 30
    elinewidth = 1
    capthick = 2
    capsize = 4
    linestyle = '-'
    marker = ['o','d', '^']

    if param == 'Voc':
        y_forw = Voc_forw_final_avg
        y_rev = Voc_rev_final_avg
        yerr_forw = Voc_forw_std
        yerr_rev = Voc_rev_std

        y_forw_rec = Voc_forw_final_avg_rec
        y_rev_rec = Voc_rev_final_avg_rec
        yerr_forw_rec = Voc_forw_std_rec
        yerr_rev_rec = Voc_rev_std_rec

        unit = ' (V)'
        ylim=[0.25,0.85]

    elif param == 'Jsc':
        y_forw = Jsc_forw_final_avg
        y_rev = Jsc_rev_final_avg
        yerr_forw = Jsc_forw_std
        yerr_rev = Jsc_rev_std

        y_forw_rec = Jsc_forw_final_avg_rec
        y_rev_rec = Jsc_rev_final_avg_rec
        yerr_forw_rec = Jsc_forw_std_rec
        yerr_rev_rec = Jsc_rev_std_rec

        unit = ' (mA/cm$^2$)'
        ylim=[-3,-29]

    elif param == 'FF':
        y_forw = FF_forw_final_avg
        y_rev = FF_rev_final_avg
        yerr_forw = FF_forw_std
        yerr_rev = FF_rev_std
```

```
        y_forw_rec = FF_forw_final_avg_rec
        y_rev_rec = FF_rev_final_avg_rec
        yerr_forw_rec = FF_forw_std_rec
        yerr_rev_rec = FF_rev_std_rec

        unit = ' (%)'
        ylim=[30,80]

    elif param == 'PCE':
        y_forw = PCE_forw_final_avg
        y_rev = PCE_rev_final_avg
        yerr_forw = PCE_forw_std
        yerr_rev = PCE_rev_std

        y_forw_rec = PCE_forw_final_avg_rec
        y_rev_rec = PCE_rev_final_avg_rec
        yerr_forw_rec = PCE_forw_std_rec
        yerr_rev_rec = PCE_rev_std_rec

        unit = ' (%)'
        ylim = [0,14]

    ax2.set_ylim(ylim)
    ax2.set_xlim([-1,25])

    ax2.minorticks_on()
    ax2.tick_params(which='major',length=5,labelsize=fontsize2)
    ax2.tick_params(which='minor',length=3,labelsize=fontsize2)

    labels_2f = ['IM/CF3BOM forw','Unpassivated forw', 'EDAI forw']
    colours_2f = ['mediumpurple','turquoise', 'orchid']
    #colours_2f = ['#4B0082', '#008B8B', '#8B008B' ]

    labels_2r = ['IM/CF3BOM rev','Unpassivated rev', 'EDAI rev']
    colours_2r = ['rebeccapurple','teal', 'mediumvioletred']
    #colours_2r = ['#B08AD7', '#00CED1', '#FF00FF']

    titles = ['Evolution of {} during continuous illumination at Voc','Evolution of {} during continuous illumination at 0V','Evolution
         of {} in the dark at 0V','Evolution of {} in the dark at 0.7V']
    if k_start == 1:
        title = titles[0]

    elif k_start == 58:
        title = titles[1]

    elif k_start == 115:
        title = titles[2]

    elif k_start == 172:
        title = titles[3]

        # labels_2f = ['Unpassivated forw'] #because Dark 0V only measured for 2PACz
        # colours_2f = ['turquoise']

        # labels_2r = ['Unpassivated rev']
        # colours_2r = ['teal']

        # marker = ['d']

    j = j_start
    k = k_start

    for i in np.arange(0,len(labels_2f),1):
        ax2.errorbar(time,y_forw[j:j+len(t_lightsoaking)],yerr=yerr_forw[j:j+len(t_lightsoaking)],elinewidth=elinewidth,capthick=capthick,capsi
```

```
            markersize = 10, linewidth = 2)
        ax2.errorbar(time,y_rev[j:j+len(t_lightsoaking)],yerr=yerr_rev[j:j+len(t_lightsoaking)],elinewidth=elinewidth,capthick=capthick,capsize
            markersize = 10, linewidth = 2)
        ax2.errorbar(time_rec,y_forw_rec[k:k+len(t_recovery)],yerr=yerr_forw_rec[k:k+len(t_recovery)],elinewidth=elinewidth,capthick=capthick,c
            markersize = 10, linewidth = 2)
        ax2.errorbar(time_rec,y_rev_rec[k:k+len(t_recovery)],yerr=yerr_rev_rec[k:k+len(t_recovery)],elinewidth=elinewidth,capthick=capthick,cap
            markersize = 10, linewidth = 2)
        j = j + len(t_lightsoaking)
        k = k + len(t_recovery)

    if j_start == 139: #to make dark 0V have a blue background everywhere
        ax2.fill_between((-5,30),-30,80,alpha=0.35,color='#a8c8ec')
        ax2.set_xlabel('Time (h)',fontsize=fontsize2)
        ax2.set_ylabel(param+unit,fontsize=fontsize2)
        ax2.legend(loc='best',fontsize=fontsize2-7, ncols = 3)
        plt.title(title.format(param),fontsize=fontsize2)
        fig2.tight_layout()
        plt.show()
    elif j_start == 208:
        ax2.fill_between((-5,30),-30,80,alpha=0.35,color='#a8c8ec')
        plt.axvline(x = 12, color = 'black', linestyle = linestyle,linewidth=0.5)
        ax2.set_xlabel('Time (h)',fontsize=fontsize2)
        ax2.set_ylabel(param+unit,fontsize=fontsize2)
        ax2.legend(loc='best',fontsize=fontsize2-7, ncols = 3)
        plt.title(title.format(param),fontsize=fontsize2)
        fig2.tight_layout()
        plt.show()
    else:
        plt.axvline(x = 12, color = 'black', linestyle = linestyle,linewidth=0.5)
        ax2.fill_between((12,30),-30,80,alpha=0.35,color='#a8c8ec')
        ax2.set_xlabel('Time (h)',fontsize=fontsize2)
        ax2.set_ylabel(param+unit,fontsize=fontsize2)
        ax2.legend(loc='upper right',fontsize=fontsize2-7, ncols = 3) ##This controls the legend settings
        plt.title(title.format(param),fontsize=fontsize2)
        fig2.tight_layout()
        plt.show()


#%%j_start and k_start determine which set of measurements is selected
j_start = [1,70,139,208] # = [Light Voc, Light 0V, Dark 0V, Dark Voc]
k_start = [1,58,115,172] # = [Light Voc, Light 0V, Dark 0V, Dark Voc]

#%% Plot lightsoaking
#j_start determines which set of measurements is selected, n = 0: Light Voc, 1: Light 0V, 2: Dark 0V, 3: Dark Voc
n = 2
plot_evolution(param='PCE',j_start=j_start[n])

#%% Plot recovery
#k_start determines which set of measurements is selected, n = 0: Light Voc, 1: Light 0V, 2: Dark 0V, 3: Dark Voc
n = 0
plot_recovery(param='Jsc',k_start=k_start[n])

#%% Plot light soaking + recovery
#j_start and k_start determine which set of measurements is selected, n = 0: Light Voc, 1: Light 0V, 2: Dark 0V, 3: Dark Voc, j and k
    should always be the same!
n = 0
plot_both(param='Jsc',j_start=j_start[n],k_start=k_start[n])

#%% Plot 4 pixel for one device
y = 'Light_Voc'
x = 2
for i in sorted(os.listdir()): #careful, if you run this twice, it will not overwrite the results but append them to the same list,
    useful for appending multiple devices to the list, if you want to wipe it, use ctrl+W
    if '_2_P' in i:
        continue
```

```
    elif i.startswith('{}_d{}_light_'.format(y,str(x).zfill(2))):
        auto.load_data(i)
        print('Plotting data from', i)
        auto.plot_JV(voltage=auto.voltage,currDens=auto.currdens,name=i,save_results=False) #use save_results=False to just plot without
            appending results to list


#%% Test

#%% Definition plotting functions
def plot_evolution(param, j_start):
    fig2 = plt.figure(figsize=(6,4),dpi=300)
    ax2 = plt.gca()

    fontsize2 = 13
    fontsize2 = 13
    elinewidth = 2
    capthick = 2
    capsize = 4
    linestyle = '-'
    marker = ['o','d']

    if param == 'Voc':
        y_forw = Voc_forw_final_avg
        y_rev = Voc_rev_final_avg
        yerr_forw = Voc_forw_std
        yerr_rev = Voc_rev_std
        unit = ' (V)'
        ylim=[0.45,0.85]

    elif param == 'Jsc':
        y_forw = Jsc_forw_final_avg
        y_rev = Jsc_rev_final_avg
        yerr_forw = Jsc_forw_std
        yerr_rev = Jsc_rev_std
        unit = ' (mA/cm$^2$)'
        ylim=[-11,-26]

    elif param == 'FF':
        y_forw = FF_forw_final_avg
        y_rev = FF_rev_final_avg
        yerr_forw = FF_forw_std
        yerr_rev = FF_rev_std
        unit = ' (%)'
        ylim=[35,75]

    elif param == 'PCE':
        y_forw = PCE_forw_final_avg
        y_rev = PCE_rev_final_avg
        yerr_forw = PCE_forw_std
        yerr_rev = PCE_rev_std
        unit = ' (%)'
        ylim = [4,13]

    ax2.set_ylim(ylim)
    #ax2.set_xlim([0,1])

    ax2.minorticks_on()
    ax2.tick_params(which='major',length=5,labelsize=fontsize2)
    ax2.tick_params(which='minor',length=3,labelsize=fontsize2)

    labels_2f = ['Salts forw','Unpassivated forw', 'EDAI']
    colours_2f = ['thistle','turquoise', 'orange']

    labels_2r = ['Salts rev','Unpassivated rev', 'EDAI']
```

```
    colours_2r = ['rebeccapurple','teal', 'orange']

    titles = ['Evolution of {} under continuous illumination at Voc','Evolution of {} under continuous illumination at 0V','Evolution
         of {} in the dark at 0V','Evolution of {} in the dark at 0.7V']
    if j_start == 1:
        title = titles[0]

    elif j_start == 70:
        title = titles[1]

    elif j_start == 139:
        title = titles[2]

    elif j_start == 208:
        title = titles[3]

        labels_2f = ['2PACz forw'] #because Dark 0V only measured for 2PACz
        colours_2f = ['turquoise']

        labels_2r = ['2PACz rev']
        colours_2r = ['teal']

        marker = ['d']

    j = j_start

    for i in np.arange(0,len(labels_2f),1):
        ax2.errorbar(time,y_forw[j:j+len(t_lightsoaking)],yerr=yerr_forw[j:j+len(t_lightsoaking)],elinewidth=elinewidth,capthick=capthick,capsi
        ax2.errorbar(time,y_rev[j:j+len(t_lightsoaking)],yerr=yerr_rev[j:j+len(t_lightsoaking)],elinewidth=elinewidth,capthick=capthick,capsize
        j = j + len(t_lightsoaking)

    ax2.set_xlabel('Time (h)',fontsize=fontsize2)
    ax2.set_ylabel(param+unit,fontsize=fontsize2)
    ax2.legend(loc='best',fontsize=fontsize2-2.5)
    plt.title(title.format(param),fontsize=fontsize2)
    fig2.tight_layout()
    plt.show()

#%%j_start and k_start determine which set of measurements is selected
j_start = [1,70,139,208] # = [Light Voc, Light 0V, Dark 0V, Dark Voc]
k_start = [1,58,115,172] # = [Light Voc, Light 0V, Dark 0V, Dark Voc]

#%% Plot lightsoaking
#j_start determines which set of measurements is selected, n = 0: Light Voc, 1: Light 0V, 2: Dark 0V, 3: Dark Voc
n = 0
plot_evolution(param='Voc',j_start=j_start[n])
```

H XRD Peak Shifting

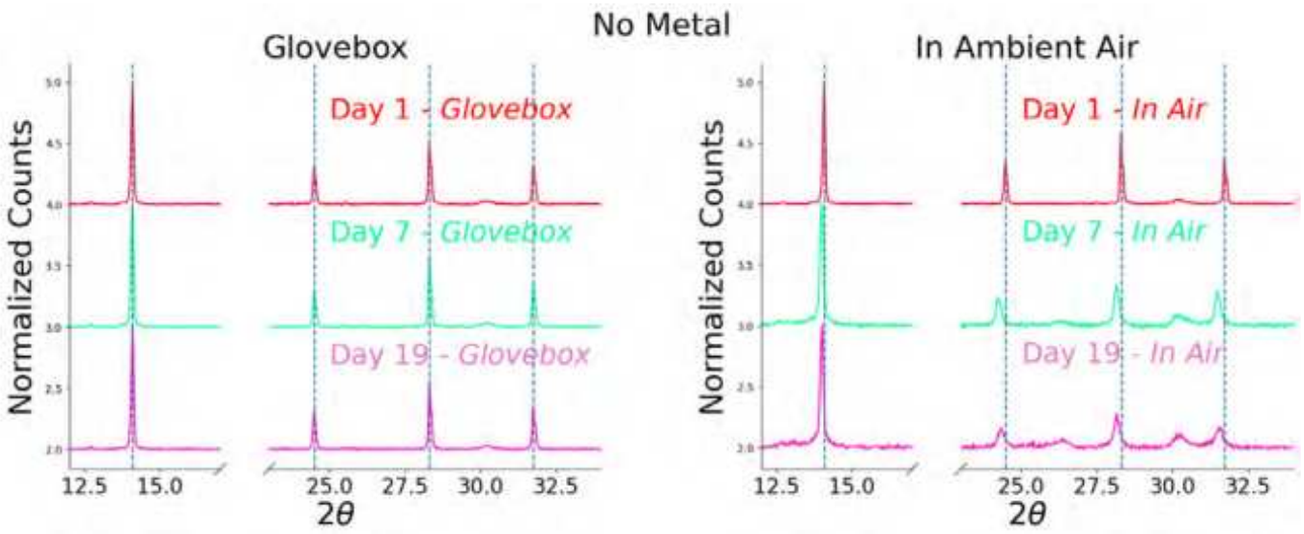


Figure 10: Peak shifting of perovskite peaks for samples stored in ambient air. An example comparing samples with an ETL and no metal contacts.