

Using deep machine learning to understand the physical performance bottlenecks in novel thin-film solar cells.

Maria Saladina and Carsten Deibel

*Institut für Physik, Technische Universität Chemnitz, 09107 Chemnitz, Germany**

Michal Krompiec

Merck Chemicals Ltd. Chilworth Technical Centre,

University Parkway, Southampton, SO16 7QD, UK and

School of Chemistry, University of Southampton Highfield, Southampton SO17 1BJ, UK[†]

Nahdia Majeed, Steve Greedy, and Roderick C. I. MacKenzie

Faculty of Engineering, University of Nottingham, Nottingham, NG7 2RD, UK[‡]

(Dated: November 9, 2019)

Abstract

There is currently a worldwide effort to develop novel materials for solar energy harvesting which are efficient, low cost and do not emit significant levels of CO_2 during manufacture. Currently when a researcher fabricates a novel device from a novel material system, it often takes many weeks of experimental effort and data analysis to understand why any given device/material combination produces an efficient or poorly optimized cell. The net result of this is that it can often take the community tens of years to transform a promising material system (e.g. perovskites/small molecule devices) to a fully optimized cell ready for production. In this work, we develop a new and rapid approach to understanding device/material performance which uses a combination of machine learning, device modeling and experiment. The method is able to provide a set of electrical device parameters (charge carrier mobilities, recombination rates etc..) in a matter of seconds, rather than days and thus offers a fast way to directly link fabrication conditions to device/material performance, pointing a way to further and more rapid optimization of light harvesting devices. We demonstrate the method by using it to understand annealing temperature and surfactant choice and in terms of charge transport and recombination constants for organic solar cells made from the P3HT:PCBM, PBTZT-stat-BDIT-8:PCBM and PTB7:PCBM material systems.

INTRODUCTION

Over the last few decades, there has been considerable academic and industrial attention focused on developing low cost and low carbon technologies to harvest the sun's energy and turn it into electricity [1]. Solar cells based on silicon currently dominate the market, however these cells require silicon of 99.9999% purity or more. Obtaining this grade of silicon from naturally occurring silicon dioxide, is a complex, expensive and multi-step process, involving liquefaction of silicon, heating to temperatures in excess 1000 K, and numerous gasification and purification steps [2]. This means, that significant amounts of energy have to be invested [2, 3] before any financial returns are seen from a silicon based PV system, furthermore it can often take years before a PV system can recoup the green house gases emitted during its manufacture [4]. For these reasons, the last two decades have seen an intense search for novel non-silicon based light harvesting materials [5].

Some of the more promising materials developed as alternatives to silicon include; semiconductors based on conducting organic molecules [6, 7], copper indium gallium selenide (CIGS) [8, 9], and more recently perovskite [10, 11] based materials, which captured the imagination of the community. Although, none of these material systems have yet become a serious competitor to silicon [12, 13], there remains an intensive and on-going effort to replace silicon, with a potentially as of yet undiscovered semiconductor [14].

Early in the development of a new material system it is not known how to optimize it for use in solar cells. This means in dozens of labs around the world, device engineers and scientists, build hundreds of thousands of candidate devices, trying to optimize the material system and boost the solar energy conversion efficiency. Translating a promising solar cell material to an optimized device can take upwards of ten years of worldwide scientific effort. Key to this development process is the need to quantify and understand the physical mechanisms behind the solar cells operation [15] which in turn dictate its efficiency. Specifically, charge carrier recombination [16], trapping [17], charge carrier mobility [18], and charge generation [19] all have to be measured throughout the cells life time and understood. However, measuring these quantities is non-trivial, because often material parameters change considerably as a function of film thickness, device structure, and deposition conditions [20]. Furthermore, many novel materials have high numbers of trap states thus changing the applied voltage bias or light intensity leads to significant changes in mobility and recombination

rates [21] making it hard to benchmark one cell/device structure/material system against another. In the more extreme case of a perovskite devices, a sea of ions within the device slowly moves as a voltage is applied fundamentally altering the device’s behavior [22], and making it yet harder to study and improve its operation.

Consequently, as the field of thin film solar technology has been developed, so has an array of measurement techniques to probe the fundamental material properties of these materials. Examples, are dark current density-voltage (JV) measurements which have been used to measure energetic disorder [23]; Charge Extraction by Linearly Increasing Voltage (CELIV), which has been used to measure effective mobility [24]; Time of Flight (ToF) photocurrent measurements for mobility and energetic disorder determination [25–27]; Transient Photo Voltage (TPV) for charge carrier lifetime measurements [28]; Transient Photo Current (TPC) again for effective mobility and energetic disorder [29]; Space-charge-limited current (SCLC) for electron and hole mobility [30]; and Impedance Spectroscopy has been used to measure recombination, mobility and contact quality [31].

In general, to extract a physical parameter from any of these measurements, one needs to fit an analytical model to the experimental data. One simple example of this approach is to use the slope of dark JV curve just after diode turnon to extract the diode ideality factor (n), which in turn can be used as a measure for the disorder within a device. One usually fits the Shockley equation,

$$J = J_0 \left(e^{\frac{qV}{nkT}} - 1.0 \right) \quad (1)$$

to the JV curve to extract this quantity. Where n is the ideality factor, V is the applied voltage, J is the diode current, J_0 is the reverse saturation current density, and all other parameters take their usual meanings. However recently Würfel et al., demonstrated that the ideality factor could also be strongly influenced by mobility [32], making it difficult to fully trust the application of equation 1. Another example would be the CELIV technique for determining charge carrier mobility. This was first demonstrated by Juška in 2000 [24], Deibel et al. then revised the method with an updated equation [33], and since then there have been other papers published either suggesting improvements on the method or questioning its validity in certain regimes [34]. In general, the drawbacks of this general approach of fitting analytical models to experimental data are four fold; 1) The most elucidating mea-

surements (i.e. ToF) can be time consuming and complex to perform; 2) analysis of the results using analytical models can be time consuming; 3) it is not always clear if the analytical models are valid for a new material system and under what experimental conditions they are valid (e.g. carrier densities, carrier gradients, light intensities etc.); and 4) often non-standard devices have to be fabricated to perform the measurements (e.g. ToF), thus there is uncertainty as to how the results relate back to a working solar cells.

Another approach to extract physical material parameters from experimental data is to fit a Monte-Carlo or finite difference based drift-diffusion device model to the data. The advantage of this approach over using analytical models, is that it can bring significant insights into device operation: one can for example examine carrier profiles across the device as a function of time or voltage. However to use these models again requires a significant investment of both human and computational time, typically fitting such a model self consistently to experimental data requires a week of computational time on a small cluster, and a few days of time from a expert in device modeling to setup and understand the simulation results (see supplementary information for a more detailed discussion of this topic). This makes these methods unsuitable for a high throughput lab environment.

The factors outlined above often prevent scientists (especially in smaller labs) from carrying out detailed analysis of their freshly fabricated devices and understanding which material parameters (mobility, recombination rates, trap densities. . .) are limiting device performance. Progress in the field is therefore slowed as scientists rarely fully understand why a device functions as it does.

In this work, we propose a new approach to linking solar cell performance to microscopic material parameters. Rather than attempting to analyze experimental data directly using analytical or numerical models, we demonstrate a method based on machine learning and deep artificial neural networks. We first use a numerical device model to generate simulated JV , and transient current/voltage curves from nominally good and nominally bad solar cells. We then use these curves to train deep artificial neural networks to identify the physical reasons for performance bottlenecks in any given device and to extract material parameters directly from the simulated data sets. We find the deep neural network is able to calculate material parameters such as mobility, number of trap states, recombination time constants and parasitic resistances from standard light and dark JV curves within a few seconds, without the need for complex time domain measurements or analysis. We then use

this new method to identify performance bottlenecks in devices made from P3HT:PCBM, PTB7:PCBM and the novel PBTZT-stat-BDTT-8:PCBM material system. The result is a general and efficient turnkey method for identifying factors limiting device performance in a lab setting, this novel method will enable scientists to more quickly optimize novel material systems and thus speed up the search for a replacement for silicon solar cells.

EXPERIMENTAL

Organic photovoltaic (OPV) devices were fabricated on pre-patterned ITO-glass substrates, the substrates were cleaned using acetone, iso-propanol, and deionized-water in an ultrasonic bath. A layer of commercially available aluminium zinc oxide (AlZnO, Nanograde) was then applied as a uniform coating by doctor blade at 40 °C. The AlZnO was then annealed at 100 °C for 10 minutes in air. The active layer material PBTZT-stat-BDTT-8:PCBM was prepared by fully dissolving it in a solution of dichlorobenzene at 30 mg.cm⁻³, this solution was then blade-coated in air to achieve a thicknesses of either of 150 nm or 350 nm, thickness values were obtained using a profilometer. A 2 minutes drying period at 60 °C on a hot plate followed to ensure removal of any residual solvent. Next 0.1 ml of PEDOT:PSS (Heraeus HTL 4083) was spread and uniformly coated by doctor blade at 70 °C. Finally, Ag (100 nm) cathodes were thermally evaporated through a shadow mask to define the cell structure. The chemical structure of PBTZT-stat-BDTT-8 is depicted in figure 1a, while the energy diagram of the device is shown in figure 1b, and the physical structure of the device is depicted in figure 1c.

Current-voltage characteristics were measured using a Keithley 2400 SMU both in the dark, and while cells were illuminated by a Newport Solar Simulator at 100 mW cm⁻². The corresponding *JV* curves are plotted as dots in figure 2. All characterization was performed in a dry-nitrogen atmosphere.

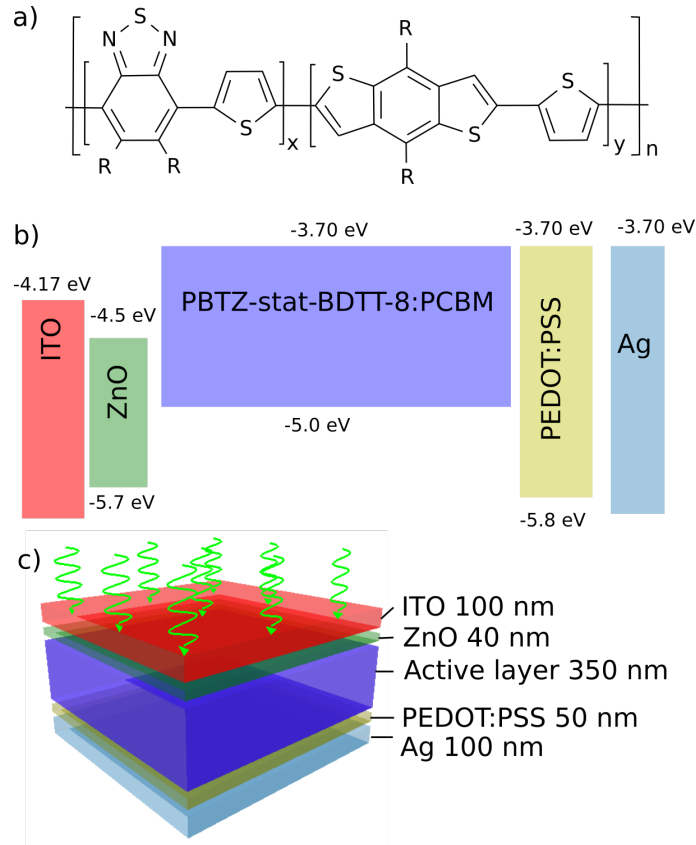


FIG. 1: a) The chemical structure of the light harvesting polymer PBTZT-stat-BDTT-8; b) The energy level diagram of the device; and c) the physical structure of the device including layer thicknesses.

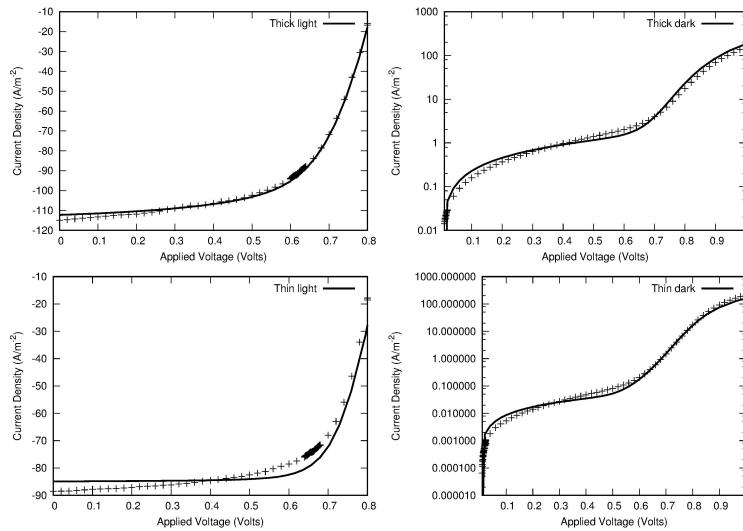


FIG. 2: Experimental JV curves measured for the thick (350 nm) device and for the thin (150 nm) device, also plotted on the same figure as lines are the fits to the experimental data using a device model. The fitting of the model to the data is discussed later.

METHOD

Over the last 20 years machine learning (ML) approaches have been increasingly used to accelerate material science discovery [35–44], applications include screening crystal structure[45], rapid searching for thermometric materials [46, 47], predicting material properties from their structure[48], predicting crystal structure [36], and screening polymers for energy harvesting applications [49, 50].

Until recently however neural networks that mimic the learning process of biological neurons have been a relatively unsuccessful class of machine learning algorithm, as they underperformed most other techniques for machine learning and data classification [51, 52]. This changed recently with two important technological developments [53]. Firstly, the increased availability of massive data sets on which the neural networks can be trained. These data sets have become available due to the exponential growth in labeled (meaning categorized by a human) images and audio data. For example, if one searches for the phrase teapot in an on-line shopping portal, one will obtain hundreds of images of 'teapots', which have all be identified as such by humans. Large data sets such as these were not available only few years ago. Secondly, to train large neural networks on these new large data sets, massively parallel computing platforms are needed, with the recent rise of Graphics processing unit (GPU) computing, and more recently dedicated neural network chips, training large networks on large data sets has become significantly easier [53].

At the heart of the artificial neural networks is the artificial neuron [54]; this is an attempt to represent a biological neuron in mathematical notation, as shown in pictorial form in figure 3b. The artificial neural network has a set of input values, $A_1 \dots A_n$ into which data is fed, and set of weights $W_1 \dots W_n$ by which the input values are multiplied. Finally the values $A_n W_n$ are summed. Thus, the value of x in figure 3b is given by

$$x = \sum_1^n A_n W_n. \tag{2}$$

The value x is then acted upon by an activation function $f(x)$, the simplest of which sets the output of the neuron: to 1 if x is above a given value, and sets the output 0 if the value of x is below a given value. In this way the neuron can make simple decisions based on the input values it receives. In this work we use the rectified linear activation function which

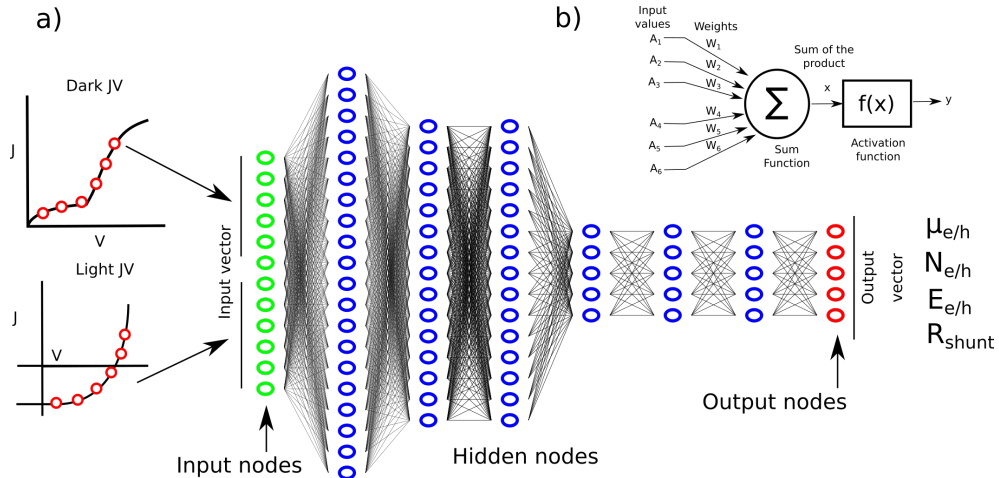


FIG. 3: A diagram of the neural network used to extract material parameters from the data within this paper, the actual network used had ten times more neurons in each hidden layer than the diagram depicts but otherwise the same structure. Visible on the left hand side of the image is the experimental (or simulated) data, with the red dots on the curves representing the points at which the curves were sampled to form input vectors for the neural network. In the diagram a light and a dark JV curve are each being sampled at 6 places to provide 12 data points to the neural network's 12 input nodes. Any number or combination of experimental measurements can be placed on the input to the network, one simply has to extend the number of input neurons, and retrain the network. The neural network itself has red input nodes, blue hidden layers and green output nodes. Each output node corresponds to a device/material parameter such as charge carrier mobility or trap density. Inset: A single neuron.

has a strong biological basis [54]. Just as in biological systems, these neurons are joined together to form networks of neurons which can be used to classify data or make complex decisions. The network we use in this work is depicted in figure 3a (to simplify the diagram only one in ten neurons in each layer is drawn). Data in this network flows from left to right. Data extracted from device measurements is presented to the network on the left hand side at the input layer (red dots). The network works on this information using its neurons (blue dots), and the estimated values of material parameters such as electron and hole mobility ($\mu_{e/h}$), number of trap states ($N_{e/h}$), or shunt resistance (R_{shunt}) are given on the output layer (green dots).

Before a neural network can be used to analyze device data it must be trained. Key to successfully training a neural network is to have a large data set. Using experimental data for training is problematic for two reasons; a) tens of thousands of JV curves measured from different devices will be needed; and b) it is often very difficult to know what the material parameters such as mobility, recombination rates really are for any given device. For

these reasons we use a Shockley-Read-Hall based drift-diffusion model (the General-purpose Photovoltaic Device Model - gpdvdm) to produce the training data [20]. We set up the device layer structure as described in the experimental section of the paper. The Highest Occupied Molecular Orbital (HOMO)/Lowest Unoccupied Molecular Orbital (LUMO) levels, and real/imaginary parts of the refractive index were manually entered into the model for each material layer. Using this base structure, we then generated a set of 20,000 devices, each with randomly assigned electrical parameters, including random recombination time constants, carrier trapping rates, trap densities, energetic disorder and parasitic resistances. For each one of these randomly generated devices we use the model to calculate the corresponding dark and light JV curve, CELIV transient, TPV transient, TPC transient at short circuit and TPC transient at reverse bias. Example curves generated by the device model are shown in figure 4.

In a lab setting a light JV and dark JV curve would almost always be measured on a newly fabricated device however, more complex time domain measurements such as CELIV or TPC would only be performed on a few selected devices to extract extra information as these measurements are more complex and time consuming to perform. By simulating a range of steady state and transient experimental measurements we will be able to choose which ones are used to train the neural network and understand which experiments provide most accurate device/material parameters.

Each JV curve or transient from a simulated measurement is effectively two columns of numbers containing one hundred to a few thousand data points. To cut down the amount of data the neural network has to process, we turn each measurement into a series of 6-12 data points. For the JV curves, this means sampling the current at $0.0 V$, $0.1 V$, $0.2 V$, \dots , $0.7 V$ and storing the values in a vector before it is fed into the neural network. The process of sampling the JV curves, and feeding the values into the neural network is shown on the left hand side of figure 3, where both a light and dark curve are being fed into the network. In a similar way any combination of time domain or steady state measurements can be fed into the neural network, one simply has to adjust the number of input nodes accordingly. We chose the number of data points to represent each measurement by balancing the need to have a small data set which is fast to train, and the need to have enough points to describe each curve accurately. Finally, we should point out that for this process to be effective, the device model must be a fair representation of a real device, the device model we use

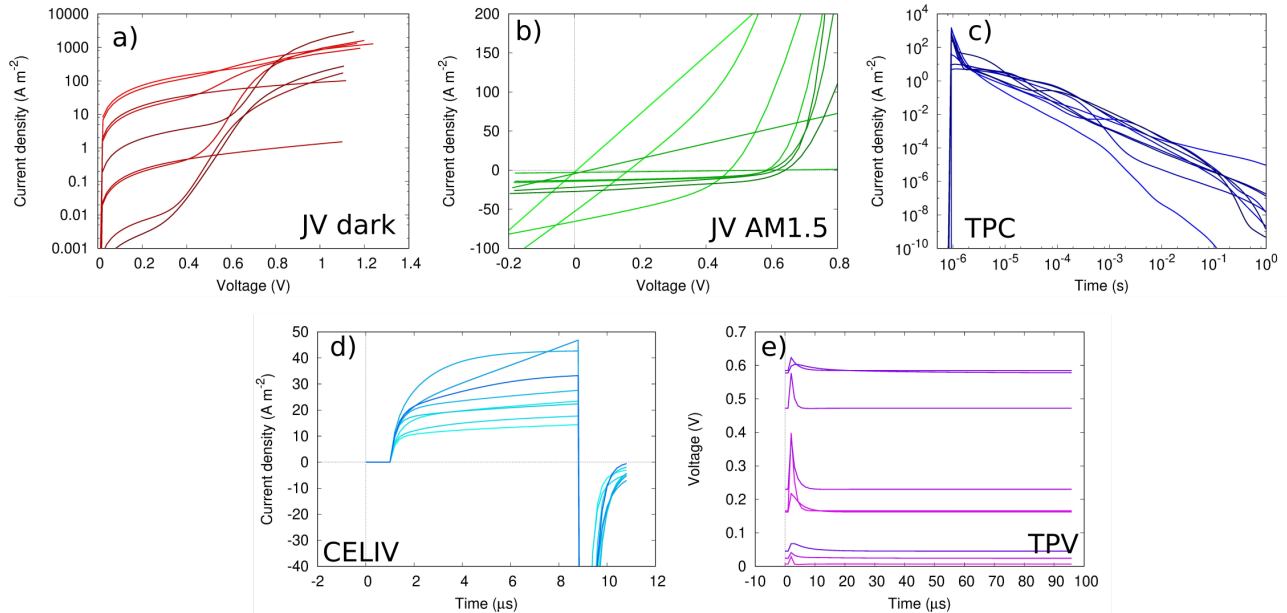


FIG. 4: Example steady state and transient curves produced by the model for training the neural network. These are a sample of 10 curves from a set of 20,000 simulated devices. For each device the structure is kept the same as in the experimental section, then random electrical parameters generated, for each device we simulate a dark JV curve (a), a light JV curve (b), a TPC transient at $-1V$, a CELIV transient and a TPV transient. The training data was split into five blocks, and in turn one of the five blocks was used for testing the network while the other five blocks were used for training. In this way every member of the data set is used for both training and testing, this approach is called five-fold cross-validation.

has previously been validated against multiple experimental data sets and proven to be predictive [20, 55–57]. More details on the implementation can be found in the SI.

The key advantage to using this Neural network/Machine learning approach is that once the neural network is trained, it can be applied to an unlimited number of similar devices with minimal computational overhead, where as with approaches base on fitting physics based device models to experimental data, significant computational resource is required each time a new device is examined. We are in effect paying the computational cost once up front rather than repeatedly at the time of use (This is discussed at length in the SI). A further advantage of our approach is that it separates the task of running device models/training neural networks from their point of use in the lab environment. One could therefore envisage an open on-line database where scientists could exchange their pre trained networks so they could be reused.

RESULTS

Table I displays how accurately the neural network can extract a given material parameter from a set of experimental data. The first column on the left displays the parameter extracted from the material system. Both electrons and holes are present in the devices, thus there will be two possible values of free carrier mobilities which could be extracted from the data. However, unless special electron or hole only devices are fabricated and measured, it will not be possible to assign a measured mobility to one carrier species. Therefore the neural network outputs three values for mobility, a maximum value (μ_{max}^{free}), a minimum value (μ_{min}^{free}) and an average value μ_{avg}^{free} . The maximum and minimum values can be assigned to either the electron or hole mobilities, but from our approach we cannot determine which one, unless we have a priori knowledge, as to which material is likely to be more mobile. Often in polymer:fullerene blends one can assume the electron transporting fullerene will be more conductive than the hole transporting polymer. Both tail slopes ($E_{e,h}^U$, where U stands for Urbach) and trap densities ($N_{e,h}$) are treated in the same way with the network outputting maximum, minimum and average values.

The value of mobility can vary by up to five orders of magnitude in an organic semiconductor device, therefore to describe the networks ability to extract mobility from any given data set, we give its accuracy using a log scale. Thus a value of 1.0 in the top three rows of the table would mean the mobility can be extracted from the experimental data within one order of magnitude, where as smaller values mean more accurate extraction. A similar approach is taken for values of trap density which can vary by up to 20 orders of magnitude in a real device and shunt resistance which can vary by up to six orders of magnitude in a device. Contact resistance and tail slope are not calculated on a log scale.

By looking at table I, it can be seen that the network extracts average and maximum free charge carrier mobility (μ_{avg}^{free} , μ_{max}^{free}) within less than half an order of magnitude from a dark/light JV curve, and over one order of magnitude from a CELIV/TPV transients. Of the measurement techniques investigated transient photocurrent is able to extract the minimum free charge carrier mobility (μ_{min}^{free}) most accurately to within half an order of magnitude. When JV_{dark} , JV_{light} , TPC_{-1V} , TPV , and $Suns - Voc$ measurements are combined as inputs to the network the values of mobility can all be extracted to within half an order of magnitude.

It might be somewhat surprising that the network finds it difficult to extract mobility from a CELIV transient, however there has been some discussion in the literature suggesting CELIV is a difficult way to measure mobility [34]. Values of tail slope (E_{min} , E_{max}) can be extracted from most measurements within around 14 meV, while values of trap density can be extracted to within about 1.5 orders of magnitude. Unsurprisingly, the network can extract the value of shunt resistance almost exactly from a dark JV curve. The final two parameters at the bottom of the table, μ_{avg}^{Pmax} , τ_{avg}^{Pmax} , represent the average mobility, and average recombination time constant at the maximum power point under AM1.5 solar radiation. It is interesting to note that the network can extract the value of τ_{avg}^{Pmax} quite reliably even from the dark JV curve alone.

	JV_{dark}	JV_{light}	$CELIV$	TPC_{-1V}	TPV	suns voc	JV_{dark} TPC_{-1V}	JV_{light} TPV	JV_{dark} TPC_{-1V}	JV_{light} TPV	Units
μ_{avg}^{eff}	0.3	0.2	1.3	0.8	1.0	0.6	0.1		0.1		$\log_{10}(m^2/Vs)$
μ_{max}^{eff}	0.3	0.3	1.3	0.8	1.0	0.6	0.2		0.2		$\log_{10}(m^2/Vs)$
μ_{min}^{eff}	1.3	0.7	0.8	0.5	0.5	0.9	0.3		0.3		$\log_{10}(m^2/Vs)$
E_{avg}^U	11.6	12.0	12.3	10.1	12.2	11.8	9.7		9.7		meV
E_{max}^U	13.7	13.9	14.1	10.7	14.3	13.7	10.4		11.0		meV
E_{min}^U	13.7	13.8	14.0	13.3	14.0	13.7	12.8		12.7		meV
N_{avg}	1.1	1.1	1.8	0.7	1.6	1.2	0.6		0.6		$\log_{10}(m^{-3})$
N_{max}	1.1	1.1	1.8	0.7	1.6	1.2	0.6		0.6		$\log_{10}(m^{-3})$
N_{min}	1.7	1.7	1.9	1.6	1.8	1.7	1.6		1.6		$\log_{10}(m^{-3})$
$R_{contact}$	4.9	3.9	1.5	9.9	10.10	10.2	3.9		3.9		Ω
R_{shunt}	0.004	0.9	0.3	1.1	1.1	0.4	0.01		0.01		$\log_{10}(\Omega)$
μ_{avg}^{Pmax}	1.6	1.4	1.7	1.0	1.8	1.6	0.7		0.7		$\log_{10}(m^2/Vs)$
τ_{avg}^{Pmax}	0.8	0.7	1.3	0.4	1.3	0.8	0.3		0.3		$\log_{10}(s^{-1})$

TABLE I: A summary of the neural network’s ability to extract material parameters from dark JV curves, light JV curves, CELIV transients, TPC transients, TPV transients and a combination of the aforementioned data sets. The numbers given are the average error the network produces when estimating a material parameter. Material parameters which can vary by multiple orders of magnitude are given on a log scale, (see the far right hand column of the table for units.) Before learning commences it is usual to initiate a neural network’s weights with random numbers, consequently there is some inherit noise in the learning process and the final quality of the neural network. Therefore, we split the data set into five parts and in turn train the network four of the five sets while testing on the remaining part. This process is performed five times, and the average presented in the table above, this is process is formally referred to as five-fold cross validation. This data corresponds to the thick device. These are RMS errors to the figure caption. True v.s. predicted curves can be found in the SI, along with detailed discussion.

Figure 5 visualizes the results from the $JV_{dark}, JV_{light}, TPC_{-1V}, TPV, Suns - V_{oc}$ column of table I. The horizontal axis represents the true value of a given device parameter, while the vertical axis represents the neural network’s prediction of the parameter, all 20,000 devices in the data set are plotted on the graphs. Ideally the plots should be a perfect diagonal, any off diagonal elements represent non-perfect estimation of device parameters. If one examines the curve for R_{shunt} , it can be seen that it is a near perfect diagonal meaning extraction of this parameter is very good. The plots for mobility (μ_{avg}, max, min), recombination constant (τ_{avg}) also lie close to the diagonal. More spread out but clustered around the diagonal are N_{avg} and N_{max} . The tail slopes seem the hardest to extract from the simulated data, $E_{max,avg}$ appear to be broadly clustered around the x/y axis, while E_{min} , forms a triangle in the upper left of the plot. Again extracting the most shallow tail slope would be expected to be most physically difficult as any current from it will be masked by the current from the trap states with the broader tail[58].

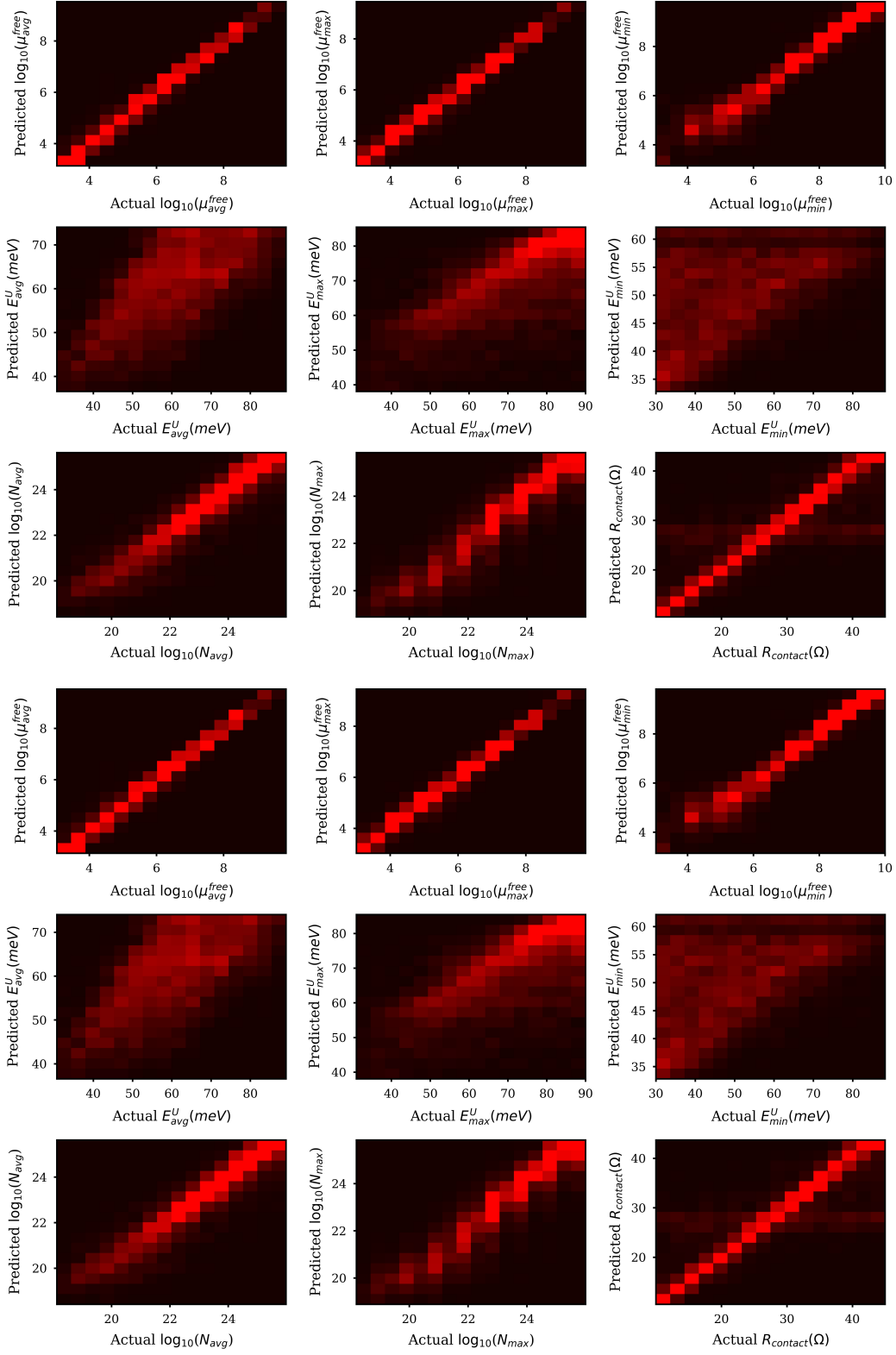


FIG. 5: A visualization of the results from the $JV_{dark}, JV_{light}, TPC_{-1V}, TPV, Suns - V_{oc}$ column of table 1. The horizontal axis represents the true value of a given device parameter, while the vertical axis represents the neural network's prediction of the parameter, all 20,000 devices in the data set are plotted on the graphs. Ideally the plots should be a perfect diagonal, any off diagonal elements represent non-perfect estimation of device parameters by the neural network.

The trained neural network was then applied to the experimental data from both the thick and thin PBTZT-stat-BDTT-8:PC₆₁BM devices shown in figure 2. The extracted material parameters are shown in Table II, it is interesting to note that the parameters extracted for both devices are quite similar as would be expected as both devices are made of the same material system, except for the values of mobility and recombination constant at P_{max} , which one would expect to be structure dependent. The minimum value for the tail slope of the trap states is 53 meV, while the maximum value is 80 meV. Although these are the first reports of energetic disorder in the PBTZT-stat-BDTT-8:PC₆₁BM material system, they compare well to previously reported values for other similar material systems. For example, the tail slopes in P3HT:PC₆₁BM have previously been reported as 35 – 40 meV for the LUMO, and 60 – 65 meV for the HOMO [29, 55]. We therefore attribute 53 meV to the electron tail slope and 80 meV to the hole tail slope.

Currently, there is only one previous report of charge carrier mobilities in the PBTZT-stat-BDTT-8:PC₆₁BM material system [30], where SCLC was used in electron and hole only devices. Values of mobility were determined of around $1 \times 10^{-7} \text{ m}^2 \text{ V}^{-1} \text{ s}^{-1}$ and of around $1 \times 10^{-8} \text{ m}^2 \text{ V}^{-1} \text{ s}^{-1}$ for electrons and holes respectively. We determine the maximum and minimum values of mobility as $9 \cdot 10^{-7} \text{ m}^2 \text{ V}^{-1} \text{ s}^{-1}$ and $4 \times 10^{-8} \text{ m}^2 \text{ V}^{-1} \text{ s}^{-1}$, while the value of effective mobility at μ_{max}^P is estimated as $3 \times 10^{-8} \text{ m}^2 \text{ V}^{-1} \text{ s}^{-1}$ for the thick device and $1 \times 10^{-6} \text{ m}^2 \text{ V}^{-1} \text{ s}^{-1}$ for the thin device. A higher mobility would be expected for a thin device, as the center of the device would be closer to the charge injecting contacts, and thus the traps would be expected to be more filled. The values of trap density (N) are given as $2 \times 10^{24} \text{ m}^{-3}$ as a maximum and $7 \times 10^{20} \text{ m}^{-3}$ as a minimum. These are comparable to the values previously given for a P3HT:PCBM material system [29]. The given value of R_{shunt} reproduces the low voltage region of the dark JV curve and thus can be considered correct although the value is low for a highly efficient device. The values of $R_{contact}$ at 40Ω is reasonable although high for an organic solar cell. The other values in the table are discussed in the next section, where we compare these results to detailed numerical modeling.

VALIDATION OF NEURAL NETWORK OUTPUT

In order to understand how accurate our neural network's values are in table II, we fit the device model used in the first half of the paper directly to the experimental data shown

Parameter	350 nm device	50 nm device	Units
μ_{avg}^{eff}	2.4×10^{-7}	2.3×10^{-7}	$m^2V^{-1}s^{-1}$
μ_{max}^{eff}	9×10^{-7}	1×10^{-7}	$m^2V^{-1}s^{-1}$
μ_{min}^{eff}	4×10^{-8}	6×10^{-8}	$m^2V^{-1}s^{-1}$
E_{avg}^U	68	69	meV
E_{max}^U	80	72	meV
E_{min}^U	53	56	meV
N_{avg}	2×10^{23}	5×10^{23}	m^{-3}
N_{max}	3×10^{23}	2×10^{24}	m^{-3}
N_{min}	7×10^{20}	1×10^{21}	m^{-3}
$R_{contact}$	44.33	43.93	Ω
R_{shunt}	3×10^4	8×10^5	Ω
μ_{avg}^{Pmax}	3×10^{-8}	1×10^{-6}	$m^2V^{-1}s^{-1}$
τ_{avg}^{Pmax}	5×10^{-5}	2×10^{-2}	s^{-1}

TABLE II: Material parameters the neural network extracted from the dark and light JV curves for thick (350nm) and thin (50nm) devices plotted in figure 2 for the PBTZT-stat-BDTP-8:PC₆₁BM material system.

in figure 2. For both the thick and thin device we use the same parameter set, but simply adjust the thickness of the active layer. It should be noted that in comparison to the neural network method presented in this paper, fitting a device model to experimental data is extremely computationally expensive, requiring a cluster of 50 CPUs. This is because to evaluate each improved guess of the parameter set the model must be rerun. The results of this fitting procedure are shown in table III.

If one compares the values in tables II and III, one can see the values are close, with mobility values being given at around $1 \times 10^{-7} m^2V^{-1}s^{-1}$, the trap density being given to around $1 \times 10^{23} m^{-3}$, the shunt resistances agree to well within an order of magnitude, the values for $R_{contact}$ disagree only by 8 Ω , but it is always hard from JV curves to distinguish which part of the resistance should be attributed to the contact and which to the internal resistance of the device. Both the recombination time constants (τ_{pmax}) and the mobility at P_{max} also agree well. Finally, the fit to the device model predicts a value of 50 meV for the tail slope, which the neural network predicts a range from 53 – 80 meV.

Simulation parameter	Extracted value
$E_{avg}^{e/h}$	50 meV
$\mu_{avg}^{e/h}$	$8 \times 10^{-8} \text{ m}^2\text{V}^{-1}\text{s}^{-1}$
$N_{avg}^{e/h}$	$8 \times 10^{23} \text{ m}^{-3}$
$R_{contact}$	36 Ω
R_{shunt}	$2.8 \times 10^4 \Omega$
μ_{pmax}	$8 \times 10^{-8} \text{ m}^2\text{V}^{-1}\text{s}^{-1}$
τ_{pmax}	$6 \times 10^{-5} \text{ s}^{-1}$

TABLE III: Model parameters needed to fit the drift-diffusion model to the experimental results from the PBTZT-stat-BDIT-8:PC₆₁BM material system shown in figure 2. The values in the table should be compared to those in table II

UNDERSTANDING THE INFLUENCE OF THERMAL ANNEALING AND ADDITIVES ON DEVICE FABRICATION.

After an organic solar cell is fabricated it is often thermally annealed, this allows the crystalline domains to grow or shrink and for molecules to rearrange their orientation. Thus thermal annealing represents a useful post fabrication method to optimize material microstructure for higher efficiency devices. Another method used to increase device efficiency is to introduce additives into the solution containing the active layer before it is deposited, these additives typically act as selective solvents for the fullerene molecules and thus alter the film formation process. Although the use of annealing and additives to improve device efficiency are well documented within the literature, it is often difficult to understand how methods change the electrical properties of the materials. To elucidate this, we fabricated four organic solar cells, two made using the P3HT:PCBM material system and two made from the PTB7:PCBM system (details of the experimental method can be found in the appendix). One of the P3HT:PCBM cells was annealed, while the other was left as cast. For the PTB7:PCBM cells one was cast with +%3 of the additive diiodooctane (DIO) in the solvent of the active layer, while the other had no DIO present. Figure 6, plots the dark JV, light JV and Suns-Voc curves for the fabricated cells.

It can be seen that the P3HT:PCBM cell which was annealed performed better than the as cast cell, while the PTB7:PCBM cell with DIO added performed better than the one without DIO. Table IV plots the device parameters extracted from the P3HT:PCBM data set using a trained neural network. The value colored green on each row represents the

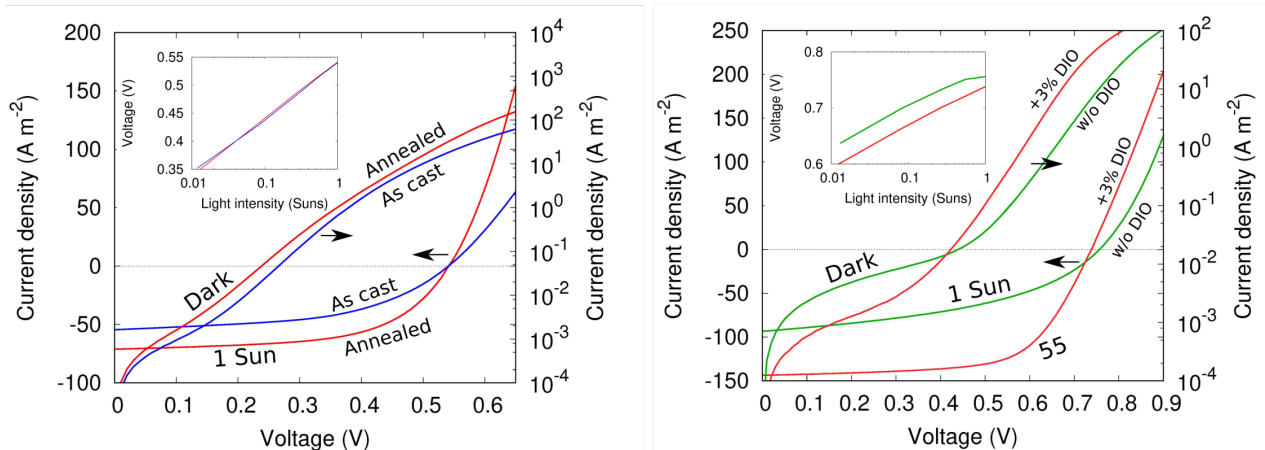


FIG. 6: Left: Dark JV , light JV , and Suns-Voc curves for annealed and as cast P3HT:PCBM cells. Right: Dark JV , light JV , and Suns-Voc curves for PTB7:PCBM cells where 3% DIO was mixed with the diiodooctane solvent when forming the active layer.

parameter which would be expected to result in a more efficient device, the one colored red represents the value which would be expected to result in a less efficient device. It can be seen that annealing has increased the free carrier mobility (μ_{avg}^{free}), unchanged the energetic disorder (E_{avg}^U), increased the mobility at the maximum power point (μ_{avg}^{Pmax}) and increased the recombination constant at the maximum power point (τ_{avg}^{Pmax}). Annealing also appears to have reduced the shunt resistance, which can also be seen from the figure directly (more current at low voltages), and increased the contact resistance.

Parameter	As cast	Annealed	Units
μ_{avg}^{eff}	2×10^{-8}	1×10^{-7}	$\text{m}^2\text{V}^{-1}\text{s}^{-1}$
E_{avg}^U	68	69	meV
N_{avg}	2×10^{25}	2×10^{25}	m^{-3}
$R_{contact}$	35	43	Ω
R_{shunt}	1.07×10^7	1.01×10^7	Ω
μ_{avg}^{Pmax}	8×10^{-9}	4×10^{-8}	$\text{m}^2\text{V}^{-1}\text{s}^{-1}$
τ_{avg}^{Pmax}	5×10^{-6}	8×10^{-6}	s^{-1}

TABLE IV: Device parameters extracted from the P3HT:PCBM devices depicted in figure 8.

Table V, plots the device parameters extracted by a neural network from the PTB7:PCBM cells depicted in figure 6. It can be seen that, the cell with the additive has a higher free carrier mobility (μ_{avg}^{free}) and a higher mobility at the maximum power point (μ_{avg}^{Pmax}). The

recombination time constant (τ_{avg}^{Pmax}) was about the same for both devices and contact resistance also slightly worse after the addition of DIO.

Parameter	w/o DIO	+3% DIO	Units
μ_{avg}^{free}	5×10^{-10}	1×10^{-9}	$\text{m}^2\text{V}^{-1}\text{s}^{-1}$
E_{avg}^U	57	69	meV
N_{avg}	3×10^{23}	2×10^{24}	m^{-3}
$R_{contact}$	38	44	Ω
R_{shunt}	1.3×10^7	1.1×10^7	Ω
μ_{avg}^{Pmax}	3×10^{-9}	1×10^{-9}	$\text{m}^2\text{V}^{-1}\text{s}^{-1}$
τ_{avg}^{Pmax}	1×10^{-4}	6×10^{-5}	s^{-1}

TABLE V: Device parameters extracted from the *PTB7 : PCBM* devices depicted in figure 8.

Finally, it should be noted that neural networks are not the only machine learning method that can be applied to this data set to extract material parameters. In the supplementary information we apply a k-nearest neighbour regression to the data set to produce a duplicate of table 1, with the results being only marginally worse than those produced by the neural network. We chose neural networks for this work because they are arguably currently producing some of the most exciting advances in machine learning.

CONCLUSION

We have demonstrated that a deep neural network can be used to quickly and robustly extract material parameters from experimental data, reducing the time for device/material parameter extraction from over a week using traditional modeling techniques to seconds. We have verified the results against a more traditional strategy of fitting a device model to an experimental data set. The advantage of using a deep neural network over more traditional approaches, such as fitting a model or analytical expression is three fold; a) once trained the network requires little interaction from the lab scientist to use; b) its use is instantaneous, meaning that values of mobility, trap density and recombination rates can all be calculated as soon as the experiment is over; and c) unlike the application of methods such as CELIV there are no assumptions made in the application of the neural network. We anticipate this work will be useful for those searching for more efficient 3rd generation solar cell materials. Although, we applied this method to solar cells within the scope of this paper, the method is equally applicable to other classes of devices such as sensors, light emitting diodes or transistors, as long as one can reliably simulate a device using a device model, a neural network can be trained to extract meaning from the experimental data.

ACKNOWLEDGMENTS

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research. The work of M.S. and C.D. has received funding from the European Unions Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 722651 (SEPOMO). RM would like to thank Thomas Gärtner (TU Wien) for many insightful discussions during the preparation of the manuscript.

* maria.saladina@physik.tu-chemnitz.de

† Michal.Krompiec@merckgroup.com

‡ eexnm12@exmail.nottingham.ac.uk

- [1] M. A. Green, Y. Hishikawa, W. Warta, E. D. Dunlop, D. H. Levi, J. Hohl-Ebinger, and A. W. HoBaillie. *Progress in Photovoltaics: Research and Applications*, 25 7 668–676.
- [2] S. Ranjan, S. Balaji, R. A. Panella, and B. E. Ydstie. *Computers and Chemical Engineering*, 2011, 35 8 1439 – 1453. Energy and Sustainability.
- [3] A. Louwen, W. G. J. H. M. van Sark, A. P. C. Faaij, and R. E. I. Schropp. *Nature Communications*, 2016, 7 13728.
- [4] L. Lu and H. Yang. *Applied Energy*, 2010, 87 12 3625 – 3631.
- [5] S. W. Tabernig, B. Daiber, T. Wang, and B. Ehrler. *Journal of Photonics for Energy*, 2018, 8 8 – 12.
- [6] A. Giovannitti, I. P. Maria, D. Hanifi, M. J. Donahue, D. Bryant, K. J. Barth, B. E. Makdah, A. Savva, D. Moia, M. Zetek, P. R. Barnes, O. G. Reid, S. Inal, G. Rumbles, G. G. Malliaras, J. Nelson, J. Rivnay, and I. McCulloch. *Chemistry of Materials*, 2018, 30 9 2945–2953.
- [7] D. Baran, N. Gasparini, A. Wadsworth, C. H. Tan, N. Wehbe, X. Song, Z. Hamid, W. Zhang, M. Neophytou, T. Kirchartz, C. J. Brabec, J. R. Durrant, and I. McCulloch. *Nature Communications*, 2018, 9 1 2059.
- [8] T. P. Weiss, S. Nishiwaki, B. Bissig, S. Buecheler, and A. N. Tiwari. *Phys. Chem. Chem. Phys.*, 2017, 19 30410–30417.
- [9] T. Walter, R. Herberholz, C. Müller, and H. W. Schock. *Journal of Applied Physics*, 1996, 80 8 4411–4420.
- [10] J. Burschka, N. Pellet, S.-J. Moon, R. Humphry-Baker, P. Gao, M. K. Nazeeruddin, and M. Grätzel. *Nature*, Jul 2013, 499 316.
- [11] P. S. Schulze, A. J. Bett, K. Winkler, A. Hinsch, S. Lee, S. Mastroianni, L. E. Mundt, M. Mundus, U. Würfel, S. W. Glunz, M. Hermle, and J. C. Goldschmidt. *ACS Applied Materials & Interfaces*, 2017, 9 36 30567–30574.
- [12] L. T. Lam, L. Branstetter, and I. L. Azevedo. *Environmental Research Letters*, 2018, 13 3 034038.
- [13] M. Saliba. *Science*, 2018, 359 6374 388–389.
- [14] D. Davies, K. Butler, A. Jackson, A. Morris, J. Frost, J. Skelton, and A. Walsh. *Chem*, 2016, 1 4 617 – 627.
- [15] L. Krückemeier, P. Kaienburg, J. Flohre, K. Bittkau, I. Zonno, B. Krogmeier, and T. Kirchartz. *Communications Physics*, 2018, 1 1 27.

- [16] G. Clemens, W. Alexander, and D. Carsten. *Advanced Electronic Materials*, 0 0 1700505.
- [17] A. Baumann, S. V ath, P. Rieder, M. C. Heiber, K. Tvingstedt, and V. Dyakonov. *The Journal of Physical Chemistry Letters*, 2015, 6 12 2350–2354.
- [18] B. Maynard, Q. Long, E. A. Schiff, M. Yang, K. Zhu, R. Kottokkaran, H. Abbas, and V. L. Dalal. *Applied Physics Letters*, 2016, 108 17 173505.
- [19] L. Yuze, Z. Fuwen, P. S. K. K., C. Jing-De, C. Wanzhu, Z. Qianqian, C. Kai, W. Yang, M. Wei, G. Feng, T. Jian-Xin, W. Chunru, Y. Wei, H. J. M., and Z. Xiaowei. *Advanced Materials*, 30 16 1706363.
- [20] H. M ackel and R. C. I. MacKenzie. *Phys. Rev. Applied*, Mar 2018, 9 034020.
- [21] R. C. I. MacKenzie, T. Kirchartz, G. F. A. Dibb, and J. Nelson. *The Journal of Physical Chemistry C*, 2011, 115 19 9806–9813.
- [22] C. Eames, J. M. Frost, P. R. F. Barnes, B. C. O’Regan, A. Walsh, and M. S. Islam. *Nature Communications*, Jun 2015, 6 7497.
- [23] T. Kirchartz, B. E. Pieters, J. Kirkpatrick, U. Rau, and J. Nelson. *Physical Review B*, 2011, 83 11 .
- [24] G. Juška, K. Arlauskas, M. Vili unas, and J. Ko cka. *Phys. Rev. Lett.*, May 2000, 84 4946–4949.
- [25] F. Steiner, S. Foster, A. Losquin, J. Labram, T. D. Anthopoulos, J. M. Frost, and J. Nelson. *Mater. Horiz.*, 2015, 2 113–119.
- [26] G. F. Seynhaeve, R. P. Barclay, G. J. Adriaenssens, and J. M. Marshall. *Phys. Rev. B*, 1989, 39 10196–10205.
- [27] C. Main, S. Reynolds, R. I. Badran, and J. M. Marshall. *Journal of Applied Physics*, 2000, 88 2 1190–1192.
- [28] S. Wheeler, D. Bryant, J. Troughton, T. Kirchartz, T. Watson, J. Nelson, and J. R. Durrant. *The Journal of Physical Chemistry C*, 2017, 121 25 13496–13506.
- [29] R. A. Street. *Phys. Rev. B*, Aug 2011, 84 075208.
- [30] I. Jeon, R. Sakai, S. Seo, G. E. Morse, H. Ueno, T. Nakagawa, Y. Qian, S. Maruyama, and Y. Matsuo. *J. Mater. Chem. A*, 2018, 6 5746–5751.
- [31] E. Ghahremanirad, A. Bou, S. Olyae, and J. Bisquert. *The Journal of Physical Chemistry Letters*, 2017, 8 7 1402–1406.
- [32] U. W urfel, D. Neher, A. Spies, and S. Albrecht. *Nature Communications*, Apr 2015, 6 6951. Article.

- [33] J. Lorrmann, B. H. Badada, O. Ingans, V. Dyakonov, and C. Deibel. *Journal of Applied Physics*, 2010, 108 11 113705.
- [34] R. Hanfland, M. A. Fischer, W. Brütting, U. Würfel, and R. C. I. MacKenzie. *Applied Physics Letters*, 2013, 103 6 063904.
- [35] A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, et al. *Apl Materials*, 2013, 1 1 011002.
- [36] C. C. Fischer, K. J. Tibbetts, D. Morgan, and G. Ceder. *Nature materials*, 2006, 5 8 641.
- [37] P. G. Boyd, Y. Lee, and B. Smit. *Nature Reviews Materials*, 2017, 2 8 17037.
- [38] S. R. Kalidindi, D. B. Brough, S. Li, A. Cecen, A. L. Blekh, F. Y. P. Congo, and C. Campbell. *Mrs Bulletin*, 2016, 41 8 596–602.
- [39] A. Jain, G. Hautier, S. P. Ong, and K. Persson. *Journal of Materials Research*, 2016, 31 8 977–994.
- [40] K. Rajan. *Materials Today*, 2005, 8 10 38 – 45.
- [41] K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev, and A. Walsh. *Nature*, 2018, 559 7715 547–555.
- [42] K. Takahashi, L. Takahashi, I. Miyazato, and Y. Tanaka. *ACS Photonics*, Mar 2018, 5 3 771–775.
- [43] K. Takahashi and Y. Tanaka. *Dalton Transactions*, 2016, 45 26 10497–10499.
- [44] Z. Ren, F. Oviedo, M. Thway, S. I. P. Tian, Y. Wang, H. Xue, J. D. Perea, M. Layurova, T. Heumueller, E. Birgersson, A. Aberle, C. J. Brabec, R. Stangl, S. Sun, F. Lin, I. M. Peters, and T. Buonassisi. 2019.
- [45] A. O. Oliynyk, L. A. Adutwum, J. J. Harynyuk, and A. Mar. *Chemistry of Materials*, 2016, 28 18 6672–6681.
- [46] M. W. Gaultois, A. O. Oliynyk, A. Mar, T. D. Sparks, G. J. Mulholland, and B. Meredig. *APL Materials*, 2016, 4 5 053213.
- [47] R. Ramprasad, R. Batra, G. Pilania, A. Mannodi-Kanakithodi, and C. Kim. *npj Computational Materials*, 2017, 3 1 54.
- [48] G. Pilania, C. Wang, X. Jiang, S. Rajasekaran, and R. Ramprasad. *Scientific reports*, 2013, 3 2810.
- [49] A. Agrawal and A. Choudhary. *Apl Materials*, 2016, 4 5 053208.

- [50] P. B. Jørgensen, M. Mesta, S. Shil, J. M. García Lastra, K. W. Jacobsen, K. S. Thygesen, and M. N. Schmidt. *The Journal of chemical physics*, 2018, 148 24 241735.
- [51] A. Raba, Y. Leroy, M. Kohlstdt, U. Wrfel, and A. S. Cordan. *IEEE Journal of Photovoltaics*, July 2017, 7 4 1098–1104.
- [52] V. Margner and M. Pechwitz. In *Proceedings of Sixth International Conference on Document Analysis and Recognition*, pages 1159–1163, 2001.
- [53] A. Krizhevsky, I. Sutskever, and G. E. Hinton. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.
- [54] R. H. R. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung. *Nature*, 2000, 405 947.
- [55] R. C. I. MacKenzie, C. G. Shuttle, M. L. Chabinye, and J. Nelson. *Advanced Energy Materials*, 2 6 662–669.
- [56] R. C. I. MacKenzie, A. Gritz, S. Greedy, E. von Hauff, and J. Nelson. *Physical Review B*, may 2014, 89 19 .
- [57] Y. Gao, R. C. I. MacKenzie, Y. Liu, B. Xu, P. H. M. van Loosdrecht, and W. Tian. *Advanced Materials Interfaces*, mar 2015, 2 4 n/a–n/a.
- [58] R. C. MacKenzie, C. G. Shuttle, G. F. Dibb, N. Treat, E. von Hauff, M. J. Robb, C. J. Hawker, M. L. Chabinye, and J. Nelson. *The Journal of Physical Chemistry C*, 2013, 117 24 12407–12414.

SUPPLEMENTARY MATERIAL

It is well known that the predictive power of any given neural network is strongly dependent upon the amount of training data available. Indeed the recent progress made in neural networks is in the main due to the increased amounts of training data available from the world wide web. Therefore to investigate the influence of training data set size on the ability of our network, we vary the number of dark JV curves our network is trained on and plot the error in its ability to extract the shunt resistance. This is shown in figure 7. It can be seen that there is a rapid drop in error from between 100-1000 devices, then there is a more gradual drop in error. Even out to 10,000 devices the error keeps dropping. Generating training steady state and transient curves for 10,000 devices took around 24 hours on a work station grade computer. We did not go above this number of devices as the return in improved recognition rate did not seem worth the additional computational overhead. One should also note that not only would an enlarged data set take more time to generate, it would consume more disk space and take longer to train the network with the data set.

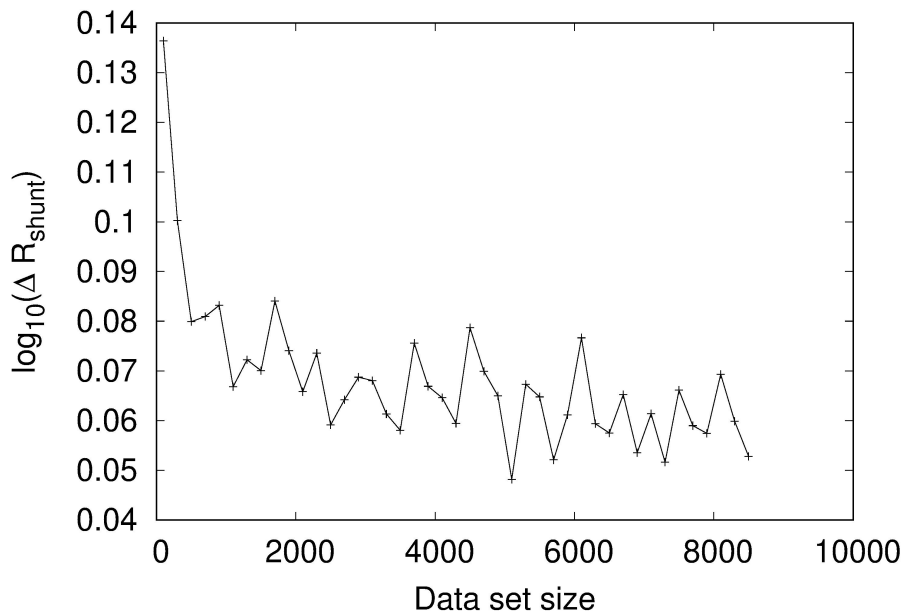


FIG. 7: Recognition rates as a function of data set and neural network size.

UNDERSTANDING THE RELATIVE DIFFERENCE BETWEEN DEVICES

When optimizing a new material system it is often less important to know the absolute values of material parameters, but rather more important to understand how the material parameters change relatively when fabrication conditions are varied. To understand how well neural network can differentiate between two slightly different devices, the network was trained on a previously published P3HT:PCBM device structure [55] for which experimental data is available [55]. The device model was then used to generate a series of dark JV curves where the mobility, density of trap states, trap tail slopes, and shunt resistances were all systematically varied one parameter at a time. The Network was then asked to extract the device parameters from the dark JV curves. The results are plotted in figure 8. For this particular device, when only using a single dark JV curve as the input data, the Network systematically overestimates free carrier mobility, underestimates the density of trap states, perfectly extracts the shunt resistance, and underestimates the change in the tail slope. However it is important to note that even with non-perfect extraction of device parameters the network is still able to tell in which direction a parameter has been changed and thus would be useful in understanding the changes in physical bottlenecks between two sets of devices.

When manually interpreting experimental data it is often hard to attribute a change in performance to any give device parameter. For example it is often hard to differentiate between an increase in free carrier mobility and a reduction in the number of trap states, both improve transport and boost device efficiency. To examine how easily the Network confuses a change in one parameter for the change in another, we vary the density of trap states over five orders of magnitude while keeping all other parameters constant. We then ask the neural network if the values of free carrier mobility and tail slope change, ideally the neural network should identify that only the density of trap states has changed. Figure 8e, plots the result. It can be seen that the Network does not confuse a change in one parameter for another.

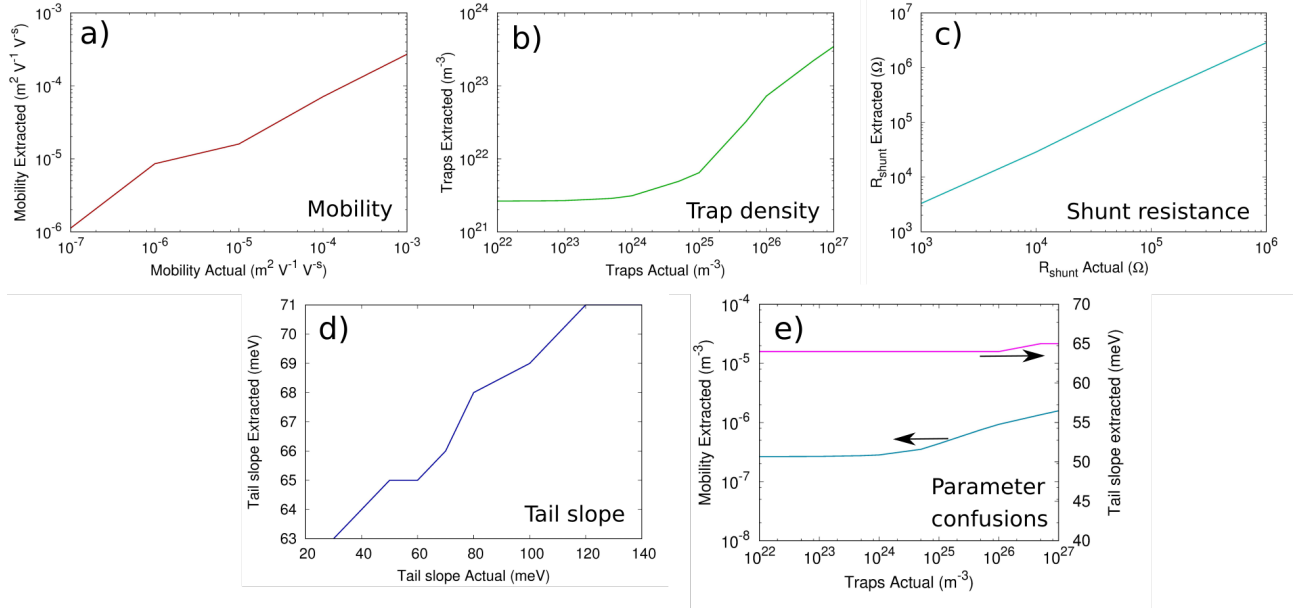


FIG. 8: In this figure a single device parameter is changed and the Network asked to extract it from a single dark JV curve. The simulations were performed for a P3HT:PCBM device [55]. Accuracy of extraction of; a) mobility; b) density of trap states; c) shunt resistance; and d) energetic tail slope. e) The ability of the neural network to differentiate between density of trap states, mobility and tail slope.

PRACTICAL ASPECTS OF IMPLEMENTATION

The data sets for learning were generated with the gpvdm (General-purpose Photovoltaic Device Model) solar cell simulation tool, this tool was developed by the authors and is available for download here <https://www.gpvdm.com>. Gpvdm was configured to generate 200 solar cells simulations a time, with each cell having a random set of device parameters. Each unique solar cell was given a random but unique 32 digit hex number to identify it. From these 200 base solar cells, sets of JV and transient measurements were produced as described in the main text of the document. After the simulations ended all but the essential files were stripped from the simulation directories using a python script to minimize disk usage and the resulting steady state/time domain curves converted into feature vectors. The 200 simulation directories were then zipped up and stored for future use. This process was repeated around 100 times until we had around 20,000 devices. This took a period of around 24 hours on a single HPC node when running in RAM disk. Gpvdm its self is written in a mixture of Python and C.

We used Googles TensorFlows (1.14) machine learning framework on a Nvidia Titan XP GPU to perform the machine learning, on an otherwise standard desktop PC. TensorFlow was controlled and driven using a series of Python 3.6 scripts.

Five-fold cross validation technique is employed to evaluate the performance of the neural network. The data set consists of 24,000 samples. The data set was split into five equal, non-overlapping, and mutually exclusive subsets. Cross validation is then performed five times. In each round, one of the five subsets becomes the test set, while the remaining samples are used to train the model. 10% of the training space is chosen as a validation set which is later used for early stopping. The pre-training samples are also randomized for each round. The scores are then averaged across the five test sets. This technique ensures that every instance in the data set gets an opportunity to participate in the test set. Also, the low variation in test scores across all folds and training on overlapping subsets on four different models indicate better performance of the model on the data the model has not been trained on.

DATA SETS

Table VI, shows which data sets were generated for each type of device

Device	Dataset size	JV Dark	JV Light	TPC -1V	TPV	CELIV	Suns - Voc
PBTZT-stat-BDTT-8 (thick)	23,411	X	X	X	X	X	X
PBTZT-stat-BDTT-8 (thin)	23,996	X	X	X	X	X	X
P3HT:PCBM (As cast)	24,085	X	X				X
P3HT:PCBM (Annealed)	24,085	X	X				X
PTB7:PCBM (w/o DIO)	23,698	X	X				X
PTB7:PCBM (+3% DIO)	24,098	X	X				X

TABLE VI: Data sets generated for this paper

WHY IS IT HARD TO FIT NUMERICAL DEVICE MODELS TO EXPERIMENTAL DATA AND WHY DO WE A DIFFERENT APPROACH TO EXTRACT DEVICE PARAMETERS?

In the main body of the text we mention that it is technically challenging to fit device models to experimental data for parameter extraction, and this is the reason device simulations are today not performed routinely by experimentalists. There are six fundamental reasons why fitting a device model self consistently to multiple sets of experimental data is a challenging proposition:

- Each time the error function, $f() = |Data_{Experimental} - Data_{Simulated}|$, is evaluated the device model must be run this usually takes between 10 seconds to either calculate a JV curve or transient curve. During a typical fit the error function must be evaluated between 1000-5000 times, thus a typical time frame for a fit is on the order of hours.
- The device equations the model solves are inherently difficult (slow) to solve as they are highly non-linear (especially when carrier traps are included), thus iterative approaches are needed to solve them. The majority of the computational overhead is in inverting the matrix containing the device equations.
- Charge carriers densities within the device can have a very wide range of values, spanning from $1 \times 10^{-3} \text{ m}^{-3}$ in very deep trap states, to $1 \times 10^{26} \text{ m}^{-3}$ near the

contact. This often results in numerical over/underflow, and thus for a given arbitrary device parameter set the device model can not be mathematically guaranteed to converge even though the device parameters may be physically reasonable. Thus $f() = |Data_{Experimental} - Data_{Simulated}|$, can often be evaluated wrongly, introducing error into the fitting process. This is a limitation of digital computers.

- The parameter space to be searched is big with parameters such as mobility, density of states, shunt resistances often varying by up to 7 orders of magnitude. There are often over 10 parameters to be searched.
- The fitting space has multiple false minima, which often require a human to spot as being unphysical, and restart the fit.
- The fitting space has multiple false minima, which often require a human to spot as being unphysical, and restart the fit.
- Due to the cost of evaluating the error function gradient decent methods are often slow, as derivatives in more than one dimension have to be evaluated.
- If one wishes to fit n experimental measurements such as JV curves and TPC curves, this will require n times more compute time.

Practically the only way to tackle this problem is to run a series of fits each starting off at different initial conditions each on a different HPC node. Then let the nodes try to find an answer for a few hours. If a few hours later a node has found a good fit, the fit process is halted, if not the fits are restarted and the process repeated. In practical terms the fits which are shown in Figure 2, took over a week to perform on a 50 core cluster. In comparison, extracting material parameters from the same data set, by first generating a fresh a training data set then training the neural network against the data took only 24 hours on a single compute node.

Furthermore, for fitting one requires a person who has a device physics/modeling background to run these simulations, often people who have this experience are not fabricating devices. Even if a group has both experimental and modeling expertise, the time it takes to perform the above described process is prohibitive and can not be performed on every device fabricated. This is why today experimentalists do not run fits as part of their every day work flow.

Once a neural network is trained for a device structure, extracting parameters requires does not require a device physics/modelling background as fitting does. A trained neural network can be applied again, and again to devices with the same structure, without the need for re-training. One could envisage an on-line library of pre-trained neural networks, from which one would select the most appropriate for a particular device structure, thus removing the need for training in the day-to-day work flow of a lab scientist all together.

It should be noted that while fitting the algorithm will spend time exploring false minima, the machine learning approach explores the whole search space, thus does not disproportionately spend time on false minima.

HYPER PARAMETERS USED FOR THE NEURAL NETWORK

Hyper parameter	Value
Optimizer	AdamOptimizer
Learning rate	0.001
Activation function	Sigmoid
Network weight initialization	xavier
Number of hidden layers	6
Number of neurons in hidden layer 1	200
Number of neurons in hidden layer 2	150
Number of neurons in hidden layer 3	50
Number of neurons in hidden layer 4	50
Number of neurons in hidden layer 5	50
Number of epochs	Early stopping with initial number of epochs set to 5000
batch size	64

THE RANGE OF SIMULATION PARAMETERS

Parameter	Min	Max	Units
μ_{avg}^{Pmax}	1×10^{-12}	$1 \times 10^{-3.0}$	$m^2/(V \cdot s)$
E_{avg}^U	30	90	meV
E_{min}^U	30	90	meV
E_{max}^U	30	90	meV
N_{avg}	1×10^{18}	1×10^{26}	m^{-3}
N_{min}	1×10^{18}	1×10^{26}	m^{-3}
N_{max}	1×10^{18}	1×10^{26}	m^{-3}
μ_{avg}^{eff}	1×10^{-10}	1×10^{-3}	$m^2/(V \cdot s)$
μ_{min}^{eff}	1×10^{-10}	1×10^{-3}	$m^2/(V \cdot s)$
μ_{max}^{eff}	1×10^{-10}	1×10^{-3}	$m^2/(V \cdot s)$
τ_{avg}^{Pmax}	1×10^{-6}	1.0	s^{-1}
R_{shunt}	1×10^3	$1 \times 10^{7.0}$	Ω
$R_{contact}$	10	45	Ω

TABLE VII: The range of simulation parameters used by the drift diffusion model.

DETAILED DESCRIPTION OF FEATURE VECTOR GENERATION

For the neural network to perform efficiently the inputs to the network should be scaled to within about an order of magnitude, usually the aim is to scale the data so that the input and output values lie between 0.0 and 1.0. Therefore before the electrical curves are turned into inputs to the neural network, they are scaled with the transforms given below in table VIII. The values of α_n , and β_n were chosen so that all values within the training set fall between 0 and 1.

As described in the main body of the text, a typical JV curve, or transient measurement contains around 1000 data points. To reduce the amount of input neurons required, to speed up the learning rate, and reduce memory consumption, we reduced the number of data points representing each JV curve or transient to around ten, by sampling each curve at the times/voltages given in the right hand column of table VIII. Furthermore, the output parameters were also scaled. Any parameter which varied by over an order of magnitude (i.e. the density of trap states or mobility) were taken as the log of the value, values containing meV were multiplied by 1000.0 and series resistance was left unchanged.

Electrical measurment	Transformation	Time/Voltage sampling intervals
JV_{dark}	$\alpha_0 \cdot \log_{10} JV_{dark} + \beta_0$	0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0 V
JV_{light}	$\alpha_1 \cdot JV_{light} + \beta_1$	0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0 V
TPC_{0V}	$\alpha_2 \cdot \log_{10} TPC_{0V} + \beta_2$	$1.1^{-6}, 2 \times 10^{-6}, 2 \times 10^{-5}, 1 \times 10^4, 0.02, 0.1, 0.2, 0.3, 0.4, 0.6, 0.7, 0.8, 0.9, 1.0, 1.2$ s
TPC_{-1V}	$\alpha_3 \cdot \log_{10} TPC_{-1V} + \beta_3$	$1.1^{-6}, 2 \times 10^{-6}, 2 \times 10^{-5}, 1 \times 10^4, 0.02, 0.1, 0.2, 0.3, 0.4, 0.6, 0.7, 0.8, 0.9, 1.0, 1.2$ s
TPV	$\alpha_4 \cdot TPV + \beta_4$	$0.14^{-6}, 0.5 \times 10^{-6}, 0.8 \times 10^{-6}, 1.0 \times 10^{-6}, 1.5 \times 10^{-6}, 2.2 \times 10^{-6}, 10 \times 10^{-6}, 20 \times 10^{-6}, 30 \times 10^{-6}, 40 \times 10^{-6}, 10^{-6}, 60 \times 10^{-6}, 50 \times, 80 \times 10^{-6}, 90 \times 10^{-6}, 100 \times 10^{-6}$ s
CELIV	$\alpha_5 \cdot CELIV + \beta_5$	$0.9 \times 10^{-6}, 1 \times 10^{-6}, 2 \times 10^{-6}, 3 \times 10^{-6}, 4 \times 10^{-6}, 5 \times 10^{-6}, 6 \times 10^{-6}, 7 \times 10^{-6}, 8 \times 10^{-6}, 9^{-6}$
$Suns - V_{oc}$	$\alpha_6 \cdot \log_{10}(Suns - V_{oc}) + \beta_6$	0.02, 0.04, 0.05, 0.1, 0.5, 0.7, 1.0

TABLE VIII: Transforming experimental data into feature vectors.

BENCHMARKING THE NEURAL NETWORK ALGORITHM AGAINST K-NN A CLUSTERING ALGORITHM

In table IX, we compare the neural network algorithm to a k-nn classifier, green corresponds to the case where the neural network outperformed a k-nn classifier, blue corresponds to both algorithms offering the same performance, and red corresponds to the case where the k-nn algorithm out performs the neural network.

	JV_{dark}	JV_{light}	$CELIV$	TPC_{-1V}	suns voc	JV_{dark} TPC_{-1V} TPV	JV_{light} TPV	JV_{dark} TPC_{-1V} suns voc	Units
μ_{avg}^{eff}	0.3/0.4	0.2/0.3	1.3/1.1	0.8/0.9	0.6/0.7	0.1/1.4		0.1/0.3	$\log_{10}(\mu)$
μ_{max}^{eff}	0.3/0.4	0.3/0.4	1.3/1.1	0.8/0.9	0.6/0.7	0.2/0.3		0.2/0.3	$\log_{10}(\mu)$
μ_{min}^{eff}	1.3/1.4	0.7/0.9	0.8/0.9	0.5/0.5	0.9/1.1	0.3/0.5		0.3/0.5	$\log_{10}(\mu)$
E_{avg}^U	11.6/12.3	12.0/12.5	12.3/12.4	10.1/9.8	11.8/12.3	9.7/-		9.7/10.7	meV
E_{max}^U	13.7/14.3	13.9/14.6	14.1/14.4	10.7/11.1	13.7/14.3	10.4/12.5		11.0/12.5	meV
E_{min}^U	13.7/14.3	13.8/14.5	14.0/14.5	13.7/13.0	13.7/14.3	12.8/13.3		12.7/13.3	meV
N_{avg}	1.1/1.4	1.1/1.3	1.8/1.8	0.7/0.9	1.2/1.3	0.6/0.9		0.6/0.9	$\log_{10}(N)$
N_{max}	1.1/1.4	1.1/1.3	1.8/1.8	0.7/0.9	1.2/1.3	0.6/0.9		0.6/0.9	$\log_{10}(N)$
N_{min}	1.7/1.8	1.7/1.8	1.9/1.9	1.6/1.7	1.7/1.8	1.6/1.7		1.6/1.7	$\log_{10}(N)$
$R_{contact}$	4.9/5.8	3.9/5.1	1.5/2.9	9.9/9.2	10.2/10.7	3.9/6.5		3.9/6.6	Ω
R_{shunt}	0.004/0.04	0.9/1.1	0.3/0.5	1.1/1.2	0.4/0.5	0.01/0.1		0.01/0.1	$\log_{10}(R_{shunt})$
μ_{avg}^{Pmax}	1.6/1.7	1.4/1.6	1.7/1.7	1.0/1.0	1.6/1.7	0.7/1.4		0.7/1.4	$\log_{10}(\mu)$
τ_{avg}^{Pmax}	0.8/1.1	0.7/0.9	1.3/1.4	0.4/0.5	0.8/1.0	0.3/0.5		0.3/0.5	$\log_{10}(\tau)$

TABLE IX: Benchmarking the neural network against a k-nn classifier, the table shows RMS errors for both algorithms. In the form RMS error Neural Network/RMS Error k-nn classifier, this was done with cross validation. Green corresponds to the case where the neural network outperformed a k-nn classifier, blue corresponds to both algorithms offering the same performance, and red corresponds to the case where the k-nn algorithm outperforms the neural network.

R2 SCORES FOR THE NEURAL NETWORK TRAINED ON THE 350nm PBTZT-STAT-BDTP-8:PC₆₁BM DEVICE.

Table X gives the R^2 values corresponding to the RMS errors in table I, values above 0.5 are highlighted in green. It can be seen that different experiments are capable of providing different material parameters reliably, for example light and dark JV curves provide significant information about the contact resistance while TPV provides little information. Physically one would expect this because TPV is an open circuit measurement unaffected by the contact resistance, while all current measured during a JV curve experiment flows through the contacts of the device, and will thus be influenced by it. It can also be seen that a wide range of experiments give reliable information on mobility and trap density but tail slope energy is the hardest to obtain information about, with only TPC at negative voltages giving an R^2 value of 0.49, this again is not surprising as tail slope energies are notoriously difficult to measure [58] due to multiple trapping/charge carrier recombination events within a standard device. As one combines multiple types of measurements (two

right hand columns), the R^2 rise to close to one for all material parameters except for the N_{min} and the values of tail slope. Again, we would expect N_{min} to be hard to measure as it's influence on current leaving the device would be swamped by the current from the N_{max} traps[58]. Finally, we note that the vales for the tail slope energy, improve considerably once multiple measurements are combined, indicating the parameter space of possible answers is being narrowed by providing multiple measurements to the neural network.

It is interesting to note that in general when the RMS errors are low one obtains good R^2 values (i.e. above 0.5).

	JV_{dark}	JV_{light}	$CELIV$	TPC_{-1V}	TPV	suns voc	JV_{dark} TPC_{-1V}	JV_{light} TPV	JV_{dark} TPC_{-1V}	JV_{light} TPV	$Suns - Voc$
μ_{avg}^{eff}	0.96	0.98	0.36	0.73	0.80	0.84	0.99		0.99		
μ_{max}^{eff}	0.96	0.97	0.34	0.72	0.83	0.86	0.99		0.99		
μ_{min}^{eff}	0.38	0.80	0.78	0.92	0.65	0.71	0.97		0.97		
E_{avg}^U	0.07	0.00	0.00	0.49	0.00	0.06	0.47		0.44		
E_{max}^U	0.00	0.00	0.00	0.01	0.00	0.07	0.0014		0.47		
E_{min}^U	0.06	0.03	0.00	0.00	0.00	0.05	0.02		0.21		
N_{avg}	0.63	0.68	0.00	0.86	0.46	0.61	0.90		0.90		
N_{max}	0.63	0.67	0.00	0.85	0.43	0.58	0.90		0.90		
N_{min}	0.15	0.17	0.00	0.24	0.00	0.13	0.23		0.23		
$R_{contact}$	0.78	0.87	0.98	0.00	0.00	0.00	0.85		0.84		
R_{shunt}	1.00	0.38	0.94	0.00	0.06	0.90	1.00		1.00		
μ_{avg}^{Pmax}	0.36	0.48	0.28	0.78	0.31	0.32	0.88		0.88		
τ_{avg}^{Pmax}	0.75	0.81	0.30	0.92	0.34	0.68	0.96		0.95		

TABLE X: R2 scores for the PBTZT-stat-BDIT-8:PC₆₁BM device.