Data-driven doping reconstruction

S. Piani[†], W. Lei[†], L. Heltai[†], N. Rotundo[‡], P. Farrell[*]
[*] Weierstrass Institute (WIAS), Mohrenstr. 39, 10117 Berlin, Germany
[†]SISSA, Via Bonomea 265, 34136 Trieste, Italy
[‡] University of Florence, Viale Morgagni 67/A 50134 Florence, Italy

Email: stefano.piani@sissa.it

***Abstract*—To reconstruct doping profiles via opto-electronic techniques (e.g. LBIC and LPS), we formulate an inverse problem based on the van Roosbroeck system. To solve it, we use neural networks fed with data created from efficient implementations of the forward model. We discuss errors of the reconstructed doping profiles as well as their robustness with respect to noise.**

I. Introduction

The non-destructive estimation of doping concentrations in semiconductor devices is of paramount importance for many applications such as crystal growth, temperature profile estimation, defect/inhomogeneity detection. Reconstructing the doping leads to an inverse problem. A number of technologies have been developed that allow the detection of doping inhomogeneities/variations via photovoltaic effects, i.e., illuminating the sample at specific positions with charge generating sources such as a laser (represented by the generation term G below), and detecting the resulting voltage or current at the contacts, e.g., LPS [1] or LBIC [2]. To reconstruct the doping profile, we generate synthetic data via the forward problem (signals generated by a laser scan along the sample's surface for a given doping), and feed them to a deep neural network. Finally, we test the robustness with respect to noise.

II. An inverse problem for the doping profile

For a known doping concentration, the *forward* problem is based on a variation of the van Roosbroeck drift-diffusion model for semi-classical transport of free electrons and holes due to a self-consistent electric field in a semiconductor. The stationary model (for boundary conditions see [1]) is given by

$$\begin{aligned} -\nabla\cdot(\varepsilon_r\nabla\psi) &= q\Big(p-n+N_D(\mathbf{x})-N_A(\mathbf{x})\Big)\\ \mp\frac{1}{q}\nabla\cdot\boldsymbol{J}_{n,p} &= G(\mathbf{x})-R, \end{aligned} \tag{1}$$

where q denotes the elementary charge, ε_r the dielectric permittivity, G the generation due to the laser beam and $R=R(\psi,\varphi_n,\varphi_p)$ the recombination. However, we assume that the doping concentration N_D-N_A takes an *unknown* profile along the x axis which we want to reconstruct, i.e., we have to solve an *inverse* problem. The current densities for electrons and holes are given by $\boldsymbol{J}_n=-q\mu_n n\nabla\varphi_n$ and $\boldsymbol{J}_p=-q\mu_p p\nabla\varphi_p$. The set of unknowns is expressed by the electrostatic potential ψ and the quasi-Fermi potentials for electrons φ_n and holes φ_p. The densities for electrons and holes are given by $n=N_c\exp[(q(\psi-\varphi_n)-E_c)/(k_BT))]$ and $p=N_v\exp[(q(\varphi_p-\psi)+E_v)/(k_BT)]$. Here, we have denoted the conduction and valence band densities of states with N_c and N_v, the Boltzmann constant with k_B and the temperature with T. Furthermore, E_c and E_v refer to the conduction and valence band-edge energies, respectively, and μ_n and μ_p to the mobilities. Due to the laser beam, we generate a measurable potential difference. That is each doping profile corresponds to a signal profile. From this relationship, we can generate a large data sets using an efficient numerical code that solves the forward problem [1] and feed these to our neural network.

III. Implementation

For every entry in our data set, we simulate the signal profile generated by a scan of the laser beam on a grid of 1,200 equispaced points in the middle of the sample (varying only in the x direction), for a known doping concentration. Every entry in our training and test sets is thus generated with 1,200 two-dimensional simulations of the forward problem (each corresponding to a different position of the laser beam). We train our network to reconstruct the map from signal to doping profile on the same grid. We build a total of 230,000 samples by picking a random number of frequencies f in the range $\{1,\dots,5\}$ and building the corresponding doping profile via:

$$N_D+N_A=C_0\left(1+\sum_{i=1}^{f}\alpha_i\sin\left(\frac{2\pi}{\lambda_i}(x+W_f(x))\right)+W_r(x)\right)$$

where $\boldsymbol{\alpha}=(\alpha_i)$ are amplitudes randomly chosen in $\{0\}\cup[0.05,0.2]$, $C_0=1.0\times10^{16}\,\mathrm{cm}^{-3}$ is a fixed average doping value, and λ_i are wavelengths randomly chosen between $10\,\mu\mathrm{m}$ and $1000\,\mu\mathrm{m}$. We use W_f and W_r as white noise sources and split our model in two parts. For the first 30,000 samples, we set $W_f=0$ and $W_r=0$, while in the last 200,000 samples, we choose some random small values for both parameters to include noise in our model.

We implemented a multilayer perceptron in `PyTorch` with 8 layers, made of a downsampler from 1,200 to 250, 2x250 Relu, 1x150 Relu, 2x100 Relu, 1x300 Relu and one upsampler from 300 to 1,200. We performed three different tests: i) (Case *clean-clean*) training and testing with clean data (30,000 samples in total, 22,500 for training and 7,500 for testing), ii) (Case *clean-noisy*) testing the same model (trained with the same data) that we had in the previous case on a noisy

978-1-6654-7899-1/22/$31.00 ©2022 IEEE

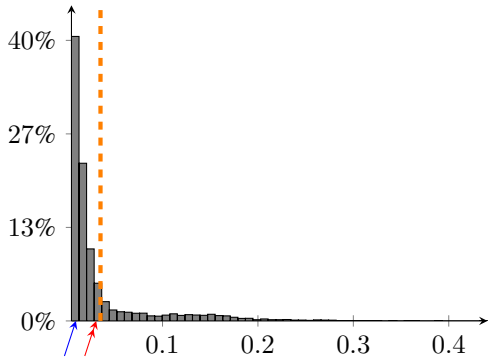


Fig. 1: The statistical distribution of the ℓ^∞ relative errors for the test case *clean-clean*. The orange line represents the average value, the blue single-headed arrow the error of the sample shown in Figure 2a which is in the 25th percentile and the red double–headed arrow the error of the the sample close to the 75th percentile and is depicted in Figure 2b.

dataset made of 7500 new samples, and iii) (Case *noisy-noisy*) both training and testing with noisy data on 200,000 samples (150,000 for training, 50,000 for testing).

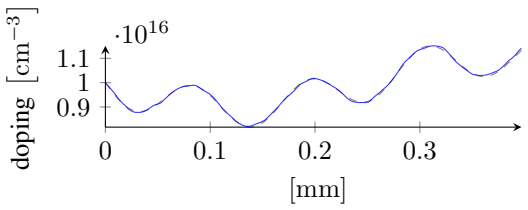
IV. Results

In Table I, we show statistical values that describe the distribution of the absolute and relative errors for the doping predictions of our model for the three different test cases described above. The error distribution of the first case is shown in Figure 1. Moreover, in Figure 2 we depict some examples of reconstructed doping profiles. The grey line is the original doping profile (only shown in x direction) and the red/blue lines are the reconstructed doping values.

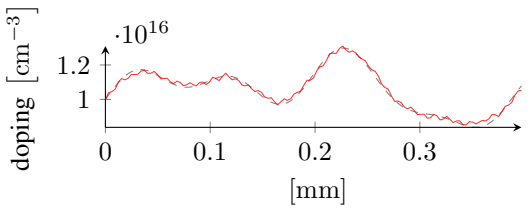
	Relative error	Absolute error (cm^{-3})
Average	3.5×10^{-2}	4.67×10^{14}
25-th percentile	9.19×10^{-3}	1.03×10^{14}
Median	1.52×10^{-2}	1.83×10^{14}
75-th percentile	3.05×10^{-2}	3.98×10^{14}
	Relative error	Absolute error (cm^{-3})
Average	1.04×10^{-1}	1.37×10^{15}
25-th percentile	3.55×10^{-2}	4.15×10^{14}
Median	7.14×10^{-2}	8.89×10^{14}
75-th percentile	1.5×10^{-1}	1.97×10^{15}
	Relative error	Absolute error (cm^{-3})
Average	4.5×10^{-2}	5.92×10^{14}
25-th percentile	1.83×10^{-2}	2.14×10^{14}
Median	2.89×10^{-2}	3.60×10^{14}
75-th percentile	4.78×10^{-2}	6.20×10^{14}

TABLE I: The ℓ^∞ relative and absolute errors for the test cases *clean-clean* (top), *clean-noisy* (mid), and *noisy-noisy* (bottom).

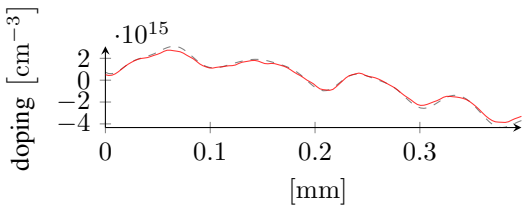
In Case *clean-noisy* (middle blocks in Table I), where we used clean training and noisy input data sets, we observe larger errors than in Case *clean-clean*. This is because we are enlarging the space of the admissible dopings without introducing new information to our model. To overcome this, we increase the number of data samples (totally 200,000 samples) and also consider noisy training sets; see the bottom



(a) 25th percentile of our distribution (case clean-clean)



(b) 75th percentile of our distribution (case clean-clean)



(c) 75th percentile of our distribution (case noisy-noisy)

Fig. 2: Some examples of predictions realized with our model (dashed line: expected value, colored line: result of the model). Figure 2a, case *clean-clean*, error within the 25th percentile, i.e. 25% of our samples are more accurate. Figure 2b case *clean-clean*, 75th percentile. Figure 2c, case *noisy-noisy*, 75 percentile.

block of Table I for the corresponding improvement of the errors. We also report the doping profile reconstruction using noisy training and input data in Figure 2 (c).

V. Summary and Outlook

We presented an inverse model for the reconstruction of the doping profile which we solved via a machine learning technique. This is relevant for many applications such as crystal growth, temperature profile estimation, defect/inhomogeneity detection. We discussed the approximation quality of our method and showed robustness with respect to noise.

References

[1] P. Farrell, S. Kayser, N. Rotundo, "Modeling and simulation of the lateral photovoltage scanning method", Computer and Mathematics with Applications, vol 102, pp. 248–260 (2021).

[2] J. Bajaj, L. O. Bubulac, P. R. Newman, W. E. Tennant, and P. M. Raccah. Spatial mapping of electrically active defects in HgCdTe using laser beam-induced current. *Journal of Vacuum Science & Technology A: Vacuum, Surfaces, and Films*, 5(5):3186–3189, sep 1987.

[3] M. Leshno, V.Y. Lin, A. Pinkus, S. Schocken, "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function" *Neural networks*, vol. 6 (1993)

[4] I. Goodfellow, Y. Bengio, A. Courville, "Deep learning" *MIT press* (2016)