Automatic TCAD Model Parameter Calibration using Autoencoder

Matthew Eng
M-PAC Lab.
San Jose State University
San Jose, USA
matthew.eng@sjsu.edu

Hiu Yung Wong*
M-PAC Lab.
San Jose State University
San Jose, USA
hiuyung.wong@sjsu.edu

***Abstract*— Modified autoencoders (AEs) have been used to capture the latent space physics of a given electrical characteristic curve (e.g. IV or CV). Therefore, it is expected that they can also be used to calibrate TCAD model parameters of novel materials such as Ga_2O_3 which is an emerging ultra-wide-bandgap (UWBG) material. In this paper, we demonstrate the use of an AE to perform automatic TCAD parameter calibration (Philips Unified Mobility model (PhuMob)) in Ga_2O_3 with 6 parameters. We also discuss a noise technique to improve calibration accuracy and an efficient training data generation method using Latin Hypercube Sampling (LHS). The machine is validated with unseen noisy curves to mimic experimental data. The PhuMob parameters extracted from the unseen curves are used in TCAD simulation and can reproduce the original curves with high accuracy (thus the calibration is successful).**

Keywords*— *Calibration, Machine Learning, Manifold Learning, Technology Computer-Aided Design (TCAD)

I. Introduction

TCAD-augmented machine learning (ML) has been proposed in recent years to solve data scarcity issues in semiconductor ML [1][2]. It has been used to perform device characteristic predictions [3]-[6], inverse design [7]- [10], and surrogate model development [9][10]. By using accurately calibrated TCAD simulations, one may also generate enough data to train a machine to troubleshoot semiconductor manufacturing defects and analyze novel materials and devices, which have been demonstrated experimentally [11]-[15]. However, TCAD simulation data can easily generate an overfitted ML model (unless substantial domain expertise is used [4][5][14]) and cannot be applied to the experiment directly due to the noise and unknown variations in experimental data. Various skills have been proposed to solve this problem [8][14][15]. Among them, autoencoder (AE) is found to be very effective [8][14]. This is because an AE can capture and learn the latent physics of any given electrical characteristic curve [3] [8][11].

Therefore, it is expected that AE can also be used to perform TCAD model parameter extraction. Since each curve is generated based on a given set of parameter values, by training an AE using curves generated with various parameters, it is expected that the AE will be able to capture the latent physics (TCAD model) of the given curve and thus can be used to predict the parameters to reproduce a given experimental curve. In this paper, the Ga_2O_3 Schottky diode forward IV is used to calibrate Ga_2O_3 Philips Unified Mobility (PhuMob) model [16] and TCAD Sentaurus [17] is used for TCAD simulations.

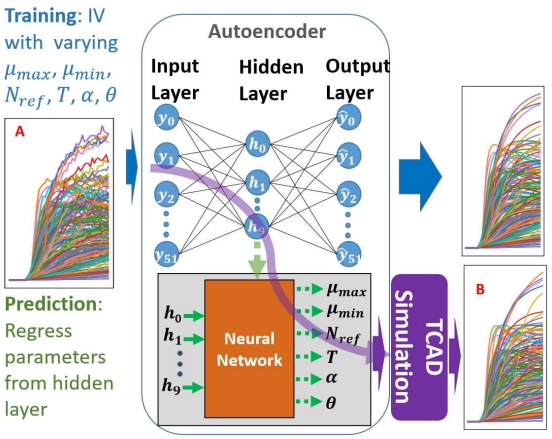

Fig. 1: Left: TCAD generated training data (A) (the split with added noise is shown, see Fig. 3). Middle: The Autoencoder framework used. Only the middle hidden layer is shown for clarity. The AE is trained using the blue path. The hidden variables are regressed against 5 PhuMob parameters and T (bottom green paths). Parameter calibration is performed using the purple path followed by TCAD simulation (B) to validate the performance.

II. Data Generation and Machine Learning

TCAD simulations are used to generate 20k forward IVs (0V to 4V) of a Ga_2O_3 Schottky diode (device structure and TCAD models can be found in [14]. For each simulation, we vary 6 parameters, the device temperature, T, and 5 PhuMob parameters (μ_{max}, μ_{min}, N_{ref}, α, and θ) (Fig. 1). The parameters for each IV are sampled within the ranges given by Table I using Latin Hypercube Sampling (LHS), which will be discussed later. Each curve is discretized to 52 points. 80% and 20% of the data are used to train the model and for validation, respectively.

The prediction model consists of two separately trained components, an undercomplete AE and a dense neural network. The AE has 9 layers, each having 52 (input layer), 128, 64, 32, 10, 32, 64, 128, and 52 (output layer) nodes,

TABLE I
Parameter Ranges Used for Dataset Generation

PARAMETER	RANGE		UNIT
	Min	*Max*	
μ_{max}	22	2000	$cm^2/(Vs)$
μ_{min}	20	1810	$cm^2/(Vs)$
N_{ref}	10^{14}	10^{20}	cm^{-3}
T	200	400	K
α	1	5	1
θ	1	5	1

*Corresponding Author: hiuyung.wong@sjsu.edu

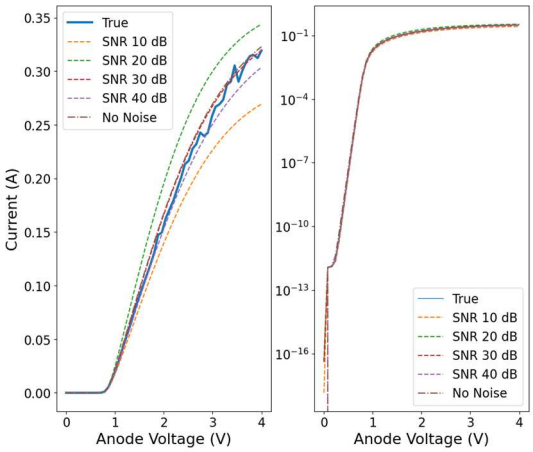

Fig. 2: True unseen noisy IV curve (emulating experimental data) vs. TCAD simulated IV using TCAD parameters extracted by the AE trained with different SNR (no noise, 10dB, 20dB, 30dB, 40dB). Left: linear-scale. Right: log-scale.

respectively. There are 10 hidden nodes in the middle layer. In principle, only 6 or fewer hidden nodes are needed to capture the 6 variations in the latent space. However, 10 are used to speed up convergence to the global minimum during the AE training (blue path in Fig. 1).

We first train the AE to reconstruct input IVs, forcing it to learn an efficient latent representation to minimize the reconstruction loss. We expect this latent representation to contain information about variable factors used to generate the IV, including the parameters we are interested in predicting. However, the learned latent space of an AE is often entangled, individual latent nodes can represent the aggregate effect of many different generative factors. Therefore, we use a second-order regression model to make parameter predictions from the AE's latent layer (green path in Fig. 1). Early results indicated that a dense neural network (with 4 hidden layers, each with 128 nodes) outperforms linear, polynomial, and K-Nearest-Neighbors regression for this application.

III. Auto-Calibration Result and Validation

To mimic the effect of non-ideal measurements in the experiment, we corrupt the unseen validation IV data by introducing additive white Gaussian noise (AWGN) with a signal-to-noise ratio (SNR) of 35 dB. Fig. 2 shows one of the unseen noisy validation curves. The curve is fed into the machine to extract the 6 parameters which are then used to perform TCAD simulations. As can be seen, the machine has extracted the PhuMob parameters and T accurately enough that the TCAD simulation IV is close to the unseen noisy data both in the linear and logarithm scale. To study the statistical accuracy of the automatic TCAD parameter calibration, 185 unseen noisy data have been tested. The R^2 of the difference between the unseen IV (curves A in Fig. 1) and the TCAD simulation IV (curves B in Fig. 1) using the extracted parameters is 0.895 (Fig 3).

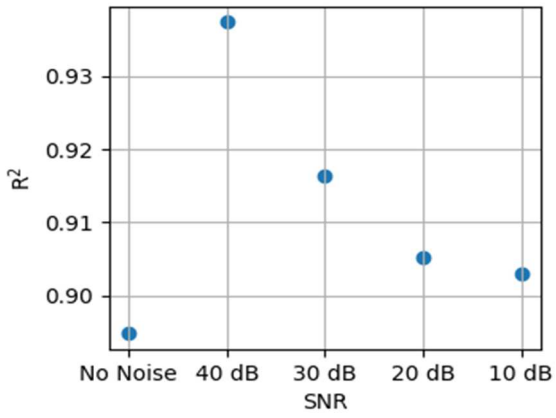

Fig. 3: R^2 score across 185 TCAD simulated IVs using model parameters calibrated by the machine trained at given SNR compared to the unseen noisy data.

IV. Accuracy Improvement with Noise Technique

To further improve the accuracy of parameter extraction, noise with different SNRs (40 dB to 10dB) is added to the training curves to reduce overfitting. For the specific example in Fig. 2, it shows that when the noise is too large (SNR = 20dB or 10dB), the prediction is worse at the high voltage level. However, statistically, adding noise improves the accuracy as shown in Fig. 3. The models trained on noisy data perform better, with the 40 dB SNR model performing best (with $R^2 \approx 0.937$, ~5% better than without noise). This robustness to noise is an important result when considering a model extension to experimental data, which often contains noise and non-ideality compared to TCAD setup (e.g. variation of WF and drift thickness).

V. Discussions

Efficient Training Data Generation - We employ LHS to generate the parameters for each IV in our simulated dataset. LHS is an efficient sampling technique shown to decrease computational effort by up to 50% compared to conventional Monte Carlo methods in some applications [18]. In our context, LHS reduces the number of simulated training data required to adequately represent the parameter space $\in \mathbb{R}^6$ (as there are 6 parameters). Fig. 4 shows the mixture discrepancy, a measure of parameter space coverage [19], as a function of sample count using both LHS and parameters drawn from independent uniform distributions. LHS produces a more uniform parameter space on average, more so when the sample count is less than 2000.

Use of Domain Expertise - The use of domain expertise can avoid the generation of unphysical data, aid interpretation of results, and improve calibration accuracy for certain regions of the parameter space. For example, from the physics point

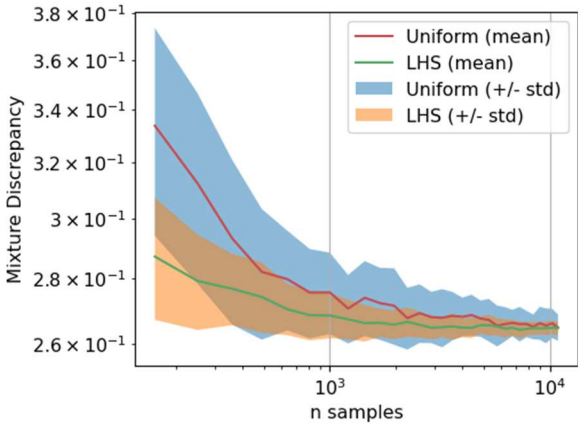

Fig. 4: Mixture discrepancy at the given number of samples across 50 trials using LHS and uniform sampling. Lower is better as the ideal score is 0.

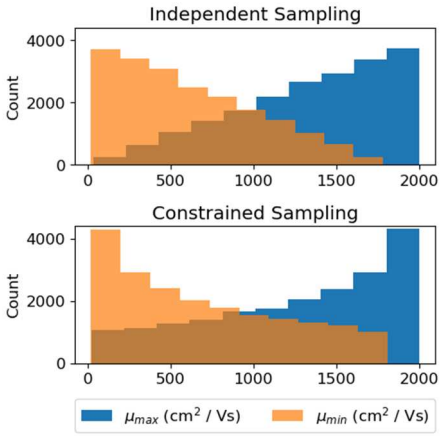

Fig. 5: Independent sampling (top) and constrained sampling (bottom) dataset histograms for μ_{max} and μ_{min}.

of view, $\mu_{max} > \mu_{min}$. One may sample μ_{max} and μ_{min} independently and then discard samples that fail this criterion (independent sampling scheme). Consequently, the dataset's μ_{max} and μ_{min} distributions skew heavily ($\frac{\mu_{max}>1800}{\mu_{max}<200} \approx 21$), as seen in Fig. 5. As such, there are relatively few training data generated with low μ_{max}, which could affect model performance on unseen data located in this underrepresented region. Instead, we sample the difference between μ_{max} and μ_{min}, given one of the values (constrained sampling scheme). This allows us to generate parameter vectors within our constraint without discarding samples, which maintains LHS coverage and produces less skew ($\frac{\mu_{max}>1800}{\mu_{max}<200} \approx 4.5$).

Accuracy of Parameters Extraction - At first glance, the true versus predicted plot for μ_{max} in Fig. 6(a) depicts poor results with $R^2 = 0.723$. The outlier and inlier histograms for these μ_{max} predictions (Fig. 6(b) and 6(c)) indicate a link between small N_{ref} values and poor predictability. To understand this result we examine the PhuMob equations [16].

The PhuMob bulk electron mobility is given by:

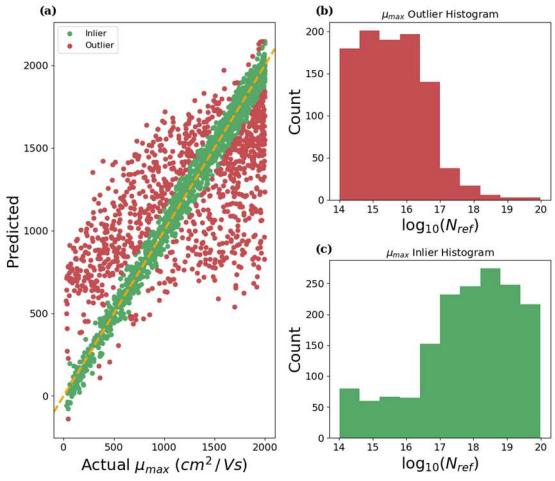

Fig. 6: (a) True vs extracted plot for μ_{max} in the 2000 validation IV. Predictions more than 150 cm^2/Vs from truth are colored red, with the remaining 'good' predictions colored green. (b) N_{ref} histogram of the outlier (red) predictions. (c) N_{ref} histogram of the inlier (green) predictions.

$$\mu_b = \left(\frac{1}{\mu_L} + \frac{1}{\mu_{DAeh}}\right)^{-1} \quad (1)$$

Given our TCAD setup, this can be approximated in terms of our parameters by:

$$\mu_b = \left(\frac{1}{\mu_{max}\left(\frac{T}{300K}\right)^{-\theta}} + \frac{1}{\left(\frac{N_{ref}}{6.0\times10^{15}}\right)^{\alpha} + \mu_c}\right)^{-1} \quad (2)$$

Where:

$$\mu_N = \frac{\mu_{max}^2}{\mu_{max} - \mu_{min}}\left(\frac{T}{300K}\right)^{3\alpha-1.5} \quad (3)$$

and,

$$\mu_c = \frac{\mu_{max}\mu_{min}}{\mu_{max} - \mu_{min}}\left(\frac{300K}{T}\right)^{0.5}. \quad (4)$$

The parameter N_{ref} varies by several orders of magnitude in the dataset and resides solely in μ_{DAeh}. When N_{ref} is large, the right-hand term in (1) tends to zero and the bulk mobility effectively equals the lattice scattering contribution, which is determined by μ_{max}. On the other hand, when N_{ref} is small, the carrier scattering dominates and μ_{max} has little to no effect on total mobility. In the latter circumstance, it may be impossible for the machine to make an accurate μ_{max} prediction. However, since μ_{max} has no effect, it does not affect the TCAD IV simulation even if the prediction is inaccurate.

To get a better idea of the model's μ_{max} prediction quality, we plot the percent error versus the evaluated partial

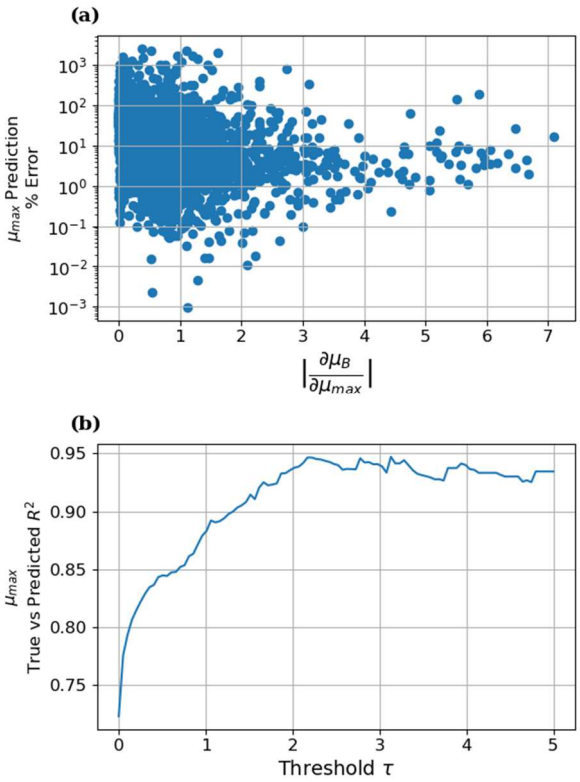

Fig. 7: (a) Partial derivative of the bulk mobility μ_B with respect to μ_{max} versus the percent error of each μ_{max} prediction in the validation set. (b) True versus predicted R^2 for μ_{max} as a function of threshold.

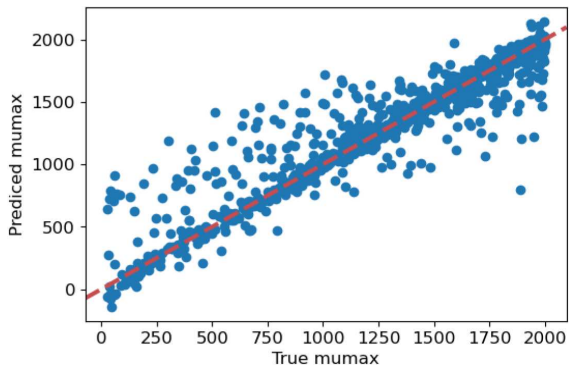

Fig. 8: True vs extracted plot for μ_{max} for $\left|\frac{\partial \mu_b}{\partial \mu_{max}}\right| > 1$.

derivative $\left|\frac{\partial \mu_b}{\partial \mu_{max}}\right|$ for each point in the validation set in Fig. 7(a). As the partial derivative approaches zero, μ_{max} contributes less to the total mobility and predictions become increasingly worse. A more accurate assessment of the model's performance is obtained by disregarding predictions where the mobility is unaffected by μ_{max} . Fig. 7(b) shows R^2 for μ_{max} as a function of a threshold τ, where data are ignored if $\left|\frac{\partial \mu_b}{\partial \mu_{max}}\right| < \tau$. With a relatively conservative threshold, $\tau = 1$, R^2 increases from 0.72 to 0.88. Fig. 8 shows the prediction of μ_{max} for $\tau = 1$ (i.e. $\left|\frac{\partial \mu_b}{\partial \mu_{max}}\right| < 1$ are ignored) and the prediction is much better than those in Fig. 7(a).

VI. Conclusions

We demonstrate the automatic TCAD calibration of 5 PhuMob parameters and *T,* with our best model achieving $R^2 \approx 0.937$ by using a modified AE trained on TCAD data. This is the first time an AE has been successfully applied to extract 6 parameters. The model is capable of calibrating noisy unseen data (emulating experimental data). It is found that by adding noise to the training data (SNR = 40dB), accuracy is improved by 5%. We implement an efficient method of parameter generation using LHS. We also find that the application of domain expertise can reduce the required size of training data and improve the performance of the machine. Furthermore, we justify our model's prediction quality of μ_{max} using the relevant PhuMob equations and propose an alternative metric.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. 2046220.

References

[1] Y. S. Bankapalli and H. Y. Wong, "TCAD Augmented Machine Learning for Semiconductor Device Failure Troubleshooting and Reverse Engineering," *2019 International Conference on Simulation of Semiconductor Processes and Devices (SISPAD)*, Udine, Italy, 2019, pp. 1-4, doi: 10.1109/SISPAD.2019.8870467.

[2] C. -W. Teo, K. L. Low, V. Narang and A. V. -Y. Thean, "TCAD-Enabled Machine Learning Defect Prediction to Accelerate Advanced Semiconductor Device Failure Analysis," 2019 International Conference on Simulation of Semiconductor Processes and Devices (SISPAD), Udine, Italy, 2019, pp. 1-4, doi: 10.1109/SISPAD.2019.8870440.

[3] K. Mehta and H. -Y. Wong, "Prediction of FinFET Current-Voltage and Capacitance-Voltage Curves Using Machine Learning With Autoencoder," in IEEE Electron Device Letters, vol. 42, no. 2, pp. 136-139, Feb. 2021, doi: 10.1109/LED.2020.3045064.

[4] J. Chen et al., "Powernet: SOI Lateral Power Device Breakdown Prediction With Deep Neural Networks," in *IEEE Access*, vol. 8, pp. 25372-25382, 2020, doi: 10.1109/ACCESS.2020.2970966.

[5] H. Carrillo-Nuñez et al., "Machine Learning Approach for Predicting the Effect of Statistical Variability in Si Junctionless Nanowire Transistors," in *IEEE Electron Device Letters*, vol. 40, no. 9, pp. 1366-1369, Sept. 2019. doi: 10.1109/LED.2019.2931839.

[6] V. Eranki, N. Yee and H. Y. Wong, "Out-of-training-range Synthetic FinFET and Inverter Data Generation using a Modified Generative Adversarial Network," in IEEE Electron Device Letters, 2022, doi: 10.1109/LED.2022.3207784.

[7] R. Wang et al., "New-Generation Design-Technology Co-Optimization (DTCO): Machine-Learning Assisted Modeling Framework," 2019 Silicon Nanoelectronics Workshop (SNW), Kyoto, Japan, 2019, pp. 1-2, doi: 10.23919/SNW.2019.8782897.

[8] K. Mehta et al., "Improvement of TCAD Augmented Machine Learning Using Autoencoder for Semiconductor Variation Identification and Inverse Design," in *IEEE Access*, vol. 8, pp. 143519-143529, 2020, doi: 10.1109/ACCESS.2020.3014470.

[9] N. Yee et al., "Rapid Inverse Design of GaN-on-GaN Diode with Guard Ring Termination for BV and (VFQ)-1 Co-Optimization," 2023 35th International Symposium on Power Semiconductor Devices and ICs (ISPSD), Hong Kong, 2023.

[10] A. Lu et al., "Vertical GaN Diode BV Maximization through Rapid TCAD Simulation and ML-enabled Surrogate Model," Solid-State Electronics, Volume 198, December 2022, 108468, https://doi.org/10.1016/j.sse.2022.108468.

[11] T. Lu et al., "Rapid MOSFET Contact Resistance Extraction From Circuit Using SPICE-Augmented Machine Learning Without Feature Extraction," in IEEE Transactions on Electron Devices, vol. 68, no. 12, pp. 6026-6032, Dec. 2021, doi: 10.1109/TED.2021.3123092.

[12] V. Eranki, T. Lu, and H. Y. Wong, "Comparison of Manifold Learning Algorithms for Rapid Circuit Defect Extraction in SPICE-Augmented Machine Learning," 2022 IEEE 19th Annual Workshop on Microelectronics and Electron Devices (WMED), 2022, pp. 1-4, doi: 10.1109/WMED55302.2022.9758032.

[13] H. Y. Wong et al., "TCAD-Machine Learning Framework for Device Variation and Operating Temperature Analysis With Experimental Demonstration," in *IEEE Journal of the Electron Devices Society*, vol. 8, pp. 992-1000, 2020, doi: 10.1109/JEDS.2020.3024669.

[14] H. Dhillon et al., "TCAD-Augmented Machine Learning With and Without Domain Expertise," in IEEE Transactions on Electron Devices, vol. 68, no. 11, pp. 5498-5503, Nov. 2021, doi: 10.1109/TED.2021.3073378.

[15] S. S. Raju et al., "Application of Noise to Avoid Overfitting in TCAD Augmented Machine Learning," 2020 International Conference on Simulation of Semiconductor Processes and Devices (SISPAD), Kobe, Japan, 2020, pp. 351-354, doi: 10.23919/SISPAD49475.2020.9241654.

[16] D. B. M. Klaassen, "A unified mobility model for device simulation—I. Model equations and concentration dependence," *Solid-State Electronics*, Vol. 35, No. 7, pp.953-959, 1992. doi: 10.1016/0038-1101(92)90325-7.

[17] Sentaurus™ Device User Guide Version S-2021.06, June. 2021.

[18] A. Olsson, G. Sandberg, and O. Dahlblom, "On Latin hypercube sampling for structural reliability analysis," Structural Safety, vol. 25, no. 1, pp. 47, 2003. https://doi.org/10.1016/S0167-4730(02)00039-5

[19] Y.-D. Zhou, et al., "Mixture discrepancy for quasi-random point sets," Journal of Complexity, vol. 29, no. 3-4, pp. 283–301, 2013.