CrossMark

Leveraging Neural Networks and Genetic Algorithms to Refine Electrode Properties in Redox Flow Batteries

Kevin M. Tenny,[1,2,*] **Richard D. Braatz,**[2,**] **Yet-Ming Chiang,**[1,3,***] and **Fikile R. Brushett**[1,2,**,z]

[1]*Joint Center for Energy Storage Research, Massachusetts Institute of Technology, Cambridge, Massachusetts, United States of America*
[2]*Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts, United States of America*
[3]*Department of Materials Science and Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts, United States of America*

Redox flow batteries are a nascent, yet promising, energy storage technology for which widespread deployment is hampered by technical and economic challenges. A performance-determining component in the reactor, present-day electrodes are often borrowed from adjacent electrochemical technologies rather than specifically designed for use in flow batteries. A lack of structural diversity in commercial offerings, coupled with the time constraints of wet-lab experiments, render broad electrode screening infeasible without a modeling complement. Herein, an experimentally validated model of a vanadium redox flow cell is used to generate polarization data for electrodes with different macrohomogeneous properties (thickness, porosity, volumetric surface area, and kinetic rate constant). Using these data sets, we then build and train a neural network with minimal average root-mean squared testing error (17.9 ± 1.8 mA cm^{-2}) to compute individual parameter sweeps along the cell polarization curve. Finally, we employ a genetic algorithm with the neural network to identify electrode property values for improving cell power density. While the developed framework does not supplant experimentation, it is generalizable to different redox chemistries and may inform future electrode design strategies.

Redox flow batteries (RFBs) have emerged as a promising technology to mitigate variations in renewable energy sources, enabling broader integration into the existing electric power system.[1,2] The wide-scale deployment of RFBs to date has been stymied by technical and economic challenges,[3,4] but advancing reactor design presents an approach to both performance enhancement and cost reduction.[5–7] While electrochemical stacks contain many constituent components that can be optimized,[8] the porous electrodes are particularly important as they facilitate a variety of performance-defining physical phenomena, including the dispersion of electrolyte,[9] the passage of current,[8] and the heterogeneous reaction site for the redox events.[10,11] The electrochemical and fluid dynamic performance of many commercial, off-the-shelf electrodes is suboptimal, as these materials are re-purposed from polymer electrolyte fuel cells[12] whose governing physics differ from RFBs. Consequently, considerable research efforts have focused on improving cell performance through on top-down, surface chemistry[10,11] and morphological[13,14] modifications of electrodes using limited vendor offerings (e.g., papers, felts, cloths). While bottom-up approaches to electrode fabrication have also been reported (e.g., electrospun[15–17]), these generated structures are typically exploratory rather than targeted. Indeed, the ideal electrode would be system-specific with proper tailored to physical and electrochemical characteristics of the redox electrolyte, as well as the geometric and fluid dynamic features of the cell and stack configuration. Unfortunately, elucidating such design criteria for a myriad electrolyte compositions and device formats is all but intractable due to present constraints in electrode manufacturability and the time- and resource-intensity of experimental structure/function screening.

Simulations, enabled by multiphysics modeling, offer an alternative domain with which to investigate flow battery design and operation. Throughout the literature, a variety of one-,[18,19] two-,[20,21]

and three-dimensional[22] RFB models have been described that contemplate electrode structures at the micro−[19] and macroscopic levels.[23] Computations at the micron scale can reveal local domain values, such as velocity profiles[24] and current distributions,[25] but these calculations are laborious and typically require reduced dimensionality or domain simplifications for improved computational tractability.[23,26] Previously, we, along with other co-authors, reported the development of an experimentally validated 3D model of a vanadium redox flow cell that was reduced to a 2D model and refined to minimize both the fluid dynamic and electrochemical discrepancies in each simulator.[27] Additionally, this work leveraged the speed of the 2D cell model to parametrically sweep a range of macrohomogeneous electrode properties bounded by experiments and first-principles modeling, which enabled us to perform 44,100 electrochemical and fluid dynamic simulations. This data set was then used to screen the design space, quantify correlative trends, and extract electrode property profiles that yielded enhanced cell performance relative to the experimental data. Importantly, however, our optimization protocol was confined to the discrete model input values due to limitations of the simulation platform that hindered a finer resolution sweep of electrode properties and the potential identification of higher-performing combinations.

Machine learning (ML) algorithms are an approach to build models with reduced computational overhead from large data sets.[28–30] While the application of ML techniques to lithium-ion batteries is an emerging research trend,[31–35] there are relatively few publications on RFBs.[28–30] We posit that this paucity stems from a lack of standardized cell designs and operating conditions, as well as limited availability of data. Therefore, we elect to use experimentally grounded simulation data sets from a single cell geometry to develop our ML algorithm. Diverse ML algorithms, including linear,[31] k-nearest neighbors,[36] random forest,[37] and neural network (NN)[28] models, have been reported in the literature, each with their own benefits and limitations. NNs are particularly popular due to their ability to describe nonlinear behavior in data sets[38] and the availability of sophisticated open-source (e.g., R and Python) and commercial (e.g., Matlab) platforms for performing the calculations.

*Electrochemical Society Student Member.
**Electrochemical Society Member.
***Electrochemical Society Fellow.
zE-mail: brushett@mit.edu

While designing the NN architecture can be time-consuming, the resulting model engenders a fast zero-dimensional function for optimization protocols that would otherwise be infeasible with high-dimensional models.

Defined by an objective function, genetic algorithms (GAs) are an optimization procedure that can estimate optimal input values[39–41] and be combined with machine learning algorithms for accelerated calculations.[42–44] This procedure iteratively solves the objective function with groups of input values and subsequently shares the values that improve the objective function output until the stopping criteria is met. While other optimization protocols exist (e.g., particle swarm[36,45]), GAs have precedence in the flow battery literature, having previously been used for parameter identification[46] and system-level monitoring.[47] As such, we chose to use GAs for an electrode analysis in vanadium redox flow cells by combining the computationally light NN with the GA to rapidly sift through a multitude of electrode features and compute the optimal electrochemical performance.

Herein, we leverage a data set of vanadium redox flow cell simulations, generated from a previously established, experimentally informed model, to validate the accuracy of predicting electrochemical performance with a NN model. We then use the NN model to perform individual parameter sweeps with other inputs held constant to isolate the effect of each on the current density across different cell potentials. Subsequently, we implement the NN model into the GA to estimate the macrohomogeneous electrode properties that predict improved power density. Although the results in this work originate from simulated data, the approach of coupling NNs to GAs can be extended to ML algorithms generated from experimental data, including a variety of redox chemistries and cell operating conditions to improve the generalizability of electrode optimization protocols. This, in turn, can enable rapid screening of a wide variety of bulk electrode morphologies that are electrolyte-specific and augment cell performance.

Methods

Data acquisition.—The data used in this work were obtained from a two-dimensional (2D) COMSOL® model reported by Cheng et al.[27] In brief, a 2D model was constructed and validated against both a three-dimensional (3D) variant and experimental data from a vanadium redox flow cell. The 2D model was subsequently used to generate a large data set by varying different electrode properties. Specifically, the electrode thickness ($H_{e,\,-}$, m), porosity (ε_-, −), volumetric surface area ($A_{V,\,-}$, m^{-1}), kinetic rate constant ($k_{0,\,-}$, m s^{-1}), and cell potential (U_{cell}, V) for the negative side of the vanadium redox flow cell were modulated and the resultant current density (i, A m^{-2}) measured. We note that the veracity of the simulated data set is largely dependent on the solver tolerance and mesh quality, whereby model discrepancies can propagate into downstream data processing. In our previous work, we performed extensive mesh validation,[27] and the variations in the multiphysics solver formulation are omitted in the subsequent data analysis. To summarize, our 2D model was discretized with 2.11×10^4 elements using a direct linear solver with the relative tolerance set to 1×10^{-4}. The raw data used for the training, validation, and testing can be found in the Supporting Information. The domain of the parametric sweep along with the parameters varied are shown in Figs. 1a and 1b, where only the negative half-cell parameters are considered; the 2D model was used to generate unique training, validation, and testing data sets that contained no overlapping input values, as shown in Table I.

Neural network development.—The NN used in this study was built following the general scheme shown in Fig. 1c, where the values of the parametric sweep and the full cell i serve as the respective input and output layers with each value considered a node. Betwixt the input and output layers are the hidden layers that connect the nodes and can be functionalized to describe nonlinear behavior within the data set.[27] The NN was formulated in Matlab
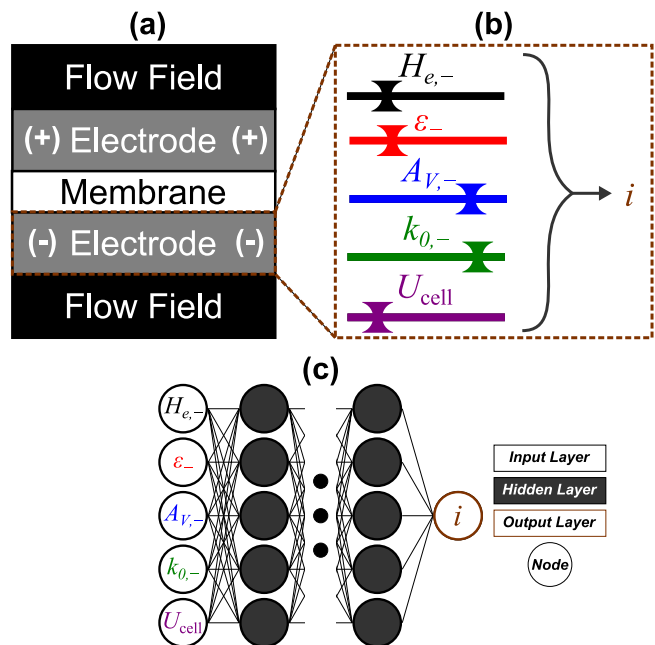

Figure 1. (a) A schematic of the simulation domain, the negative electrode of the vanadium redox flow cell. (b) The parameters $H_{e,-}$, ε_-, $A_{V,-}$, $k_{0,-}$, and U_{cell} from the negative side are adjusted to observe i of the full cell. (c) A generic neural network depicting the input layer as the parameter values and the output layer as i. Connecting the input and the output are hidden layers that, when affixed with nonlinear activation functions, can describe inherent trends within the modeling domain.

R2020a® using the training, validation, and testing data sets described in Table I, with the time required to generate each listed in the final row using the computational resources described below. The baseline NN architecture consisted of five input nodes densely attached to two hidden layers before connecting to the output layer. Each layer consisted of a bias and an activation function; the selection of hidden layer activations was determined via the training procedure (*vide infra*), and the output layer used a linear activation function. The software used the *FeedForwardNet* function with properties as described in Table SI (available online at stacks.iop.org/JES/168/050547/mmedia) of the Supporting Information. Prior to NN training, the base-10 logarithm of the data was normalized by subtracting the training data mean and dividing by the training data standard deviation. The general normalization procedure is defined as

$$(\tilde{\cdot}) = \frac{(\cdot) - \mathrm{mean}(\cdot)_{\mathrm{train}}}{\mathrm{std}(\cdot)_{\mathrm{train}}}, \qquad [1]$$

where $(\tilde{\cdot})$ represents the normalized value of either the inputs or the outputs. This normalization is carried out with respect to the training data, enabling consistent normalization for future data sets. To determine the NN architecture, the number of nodes in the two hidden layers was varied from one to five in increments of one to calculate the minimum root-mean squared error (RMSE) for the validation data using an early-stopping regularization,[48] as calculated by

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(i_{\mathrm{predicted}} - i_{\mathrm{actual}})^2}, \qquad [2]$$

where N are the total number of observations (−), i_{actual} is the current density from the testing data (A m^{-2}), and $i_{\mathrm{predicted}}$ is the current density returned from the NN (A m^{-2}). The total training lasted ca. 1 h running on an Intel® Core™ i7-9750H CPU @ 2.6 GHz, 2601 MHz, 6 Cores, 12 logical processors while employing the *Parallel Toolbox* in Matlab R2020a®.

Table I. The training, validation, and testing data generated with the 2D COMSOL® model.[27]

Input	Units	Training	Validation	Testing
$H_{e,-}$	10^{-4} m	3, 5, 10, 15	2, 6, 12	4, 8, 14
ε_-	–	0.6, 0.7, 0.8, 0.9	0.68, 0.78, 0.88	0.62, 0.72, 0.82
$A_{V,-}$	10^5 m^{-1}	1.9, 2.8, 3.7, 4.5, 4.8	3.0, 3.5, 4.0, 4.6	2.0, 2.4, 3.6, 4.3
$k_{0,-}$	10^{-7} m s^{-1}	0.2, 0.7, 1.2, 1.7, 2.2	1.1, 1.5, 1.9	1.0, 1.4, 2.1
U_{cell}	V	0.65, 0.7, 0.9, 1.1, 1.3	0.675, 0.80, 1.20	0.68, 0.85, 1.05
Approximate Acquisition Time (h)		30	3.5	3.5

Individual parameter sweeps.—Individual parameter sweeps were performed using the best-trained NN to observe how the system polarization changes by sweeping either $H_{e,-}$, ε_-, $A_{V,-}$, or $k_{0,-}$, with the rest of the parameters fixed at their maximum or minimum values. The parameter to be swept, along with U_{cell}, were changed from their global minimum to maximum value in 201 equally spaced intervals, resulting in eight sets of 40,401 simulations that required ca. 16 s in total to complete with the aforementioned computational resources (*vide supra*).

Genetic algorithm.—A schematic of GA is in Fig. 2, depicting the flow of the algorithm throughout optimization. The *GA* function in Matlab R2020a® was used to execute the protocol, where a population of 100 input vectors each with the five input values was randomly instantiated and passed into the trained NN with the fitness function being to maximize the power density (P, W m^{-2}), the product of i and U_{cell}. The data were sorted by decreasing power density, and the top 10% of data were chosen to share 10% of their values with the next population. The remaining balance were allocated random values, and the new population received a small perturbation from a Gaussian distribution that had a zero mean and standard deviation equal to that of previous generation. All additional auxiliary parameters are in Table SII of the Supporting Information.

Results and Discussion

Neural network-based model construction.—The training, validation, and testing data sets represent distinct input-output relationships used for building and assessing the NN model. In particular, i for each of the data sets spans multiple orders of magnitude as evinced by the histograms shown in Fig. 3. The large range is primarily due to the variation in U_{cell} that affects i in constructing a full polarization curve. Similar values of i are both within and between each data set, with the frequency based on the number of input parameter combinations to the system. Additionally, each of the input parameters for these data sets have variable Kendall τ rank correlative strengths with respect to i (Table II), highlighting the variable relationships in the data. The inherent nonlinearities necessitate a model representation that can describe the same nonlinear input-output behavior of the real system when regressing i quantities. We note that the generated data are specific to the interdigitated flow field cell geometry[18] and the electrode property values only reflect the negative (V$^{2+/3+}$) half of a full vanadium redox flow cell. Additionally, other performance-altering system parameters (e.g., vanadium concentrations, flow rates, tortuosity, temperature, etc.) are not incorporated in this data set to measure their influence on i. Importantly, these results do not represent an apex in vanadium flow cell performance but rather an estimated improvement at the previously specified conditions. Overall, this flexible and generic approach can be expanded to incorporate other electrode and operating conditions.

When training the NN, its architecture must be specified, which involves selecting (i) the number of nodes in each hidden layer, and (ii) the type of activation function, where both are adjusted to minimize the RMSE for the validation data. As represented, the
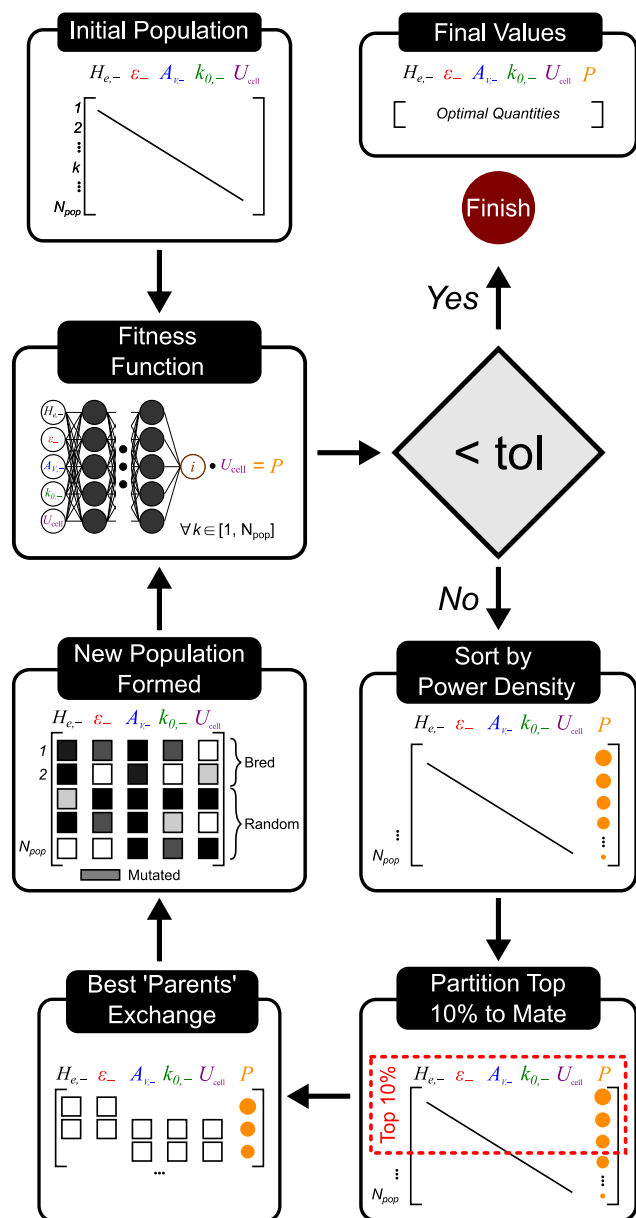

Figure 2. The genetic algorithm begins with an initial population of 100 randomized inputs that are passed into an objective function seeking to maximize P. The results are sorted in decreasing performance, and the top 10% of the population "mates" to produce the next population by sharing 10% of the input vectors. The progeny is then perturbed with a random number generated from a Gaussian distribution with a standard deviation from the previous generation. The fitness of the new population is then evaluated and repeated for n generations until the convergence criterion is achieved.
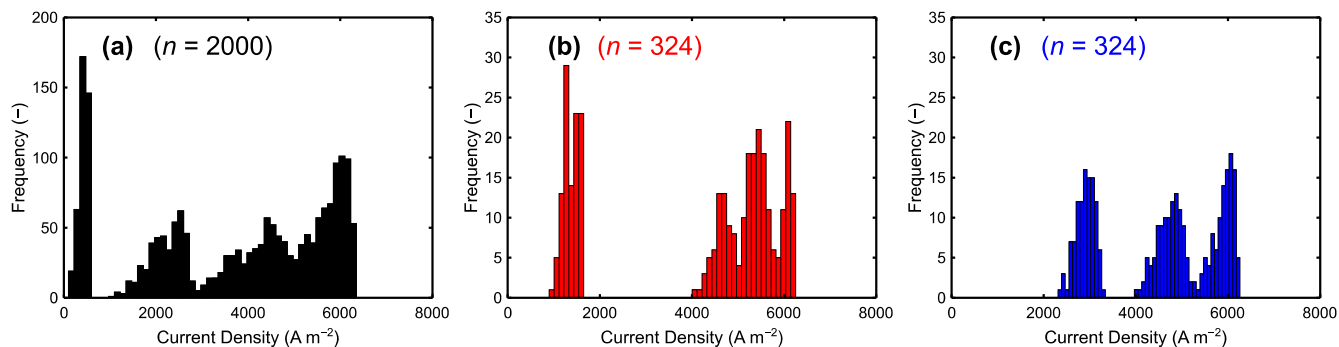
Figure 3. The distribution of the current densities from the (a) training, (b) validation, and (c) testing data sets.

Table II. The Kendall τ rank correlation coefficient for each input parameter with respect to *i*.

Correlations with *i* Data Set	Parameter				
	$H_{e,-}$	ε_-	$A_{V,-}$	$k_{0,-}$	U_{cell}
Training	0.04	−0.13	0.08	0.11	−0.83
Validation	0.17	−0.19	0.09	0.06	−0.78
Testing	0.06	−0.14	0.16	0.09	−0.82

Table III. The RMSE for the three activation functions across all data sets for five input nodes and four hidden layer nodes.

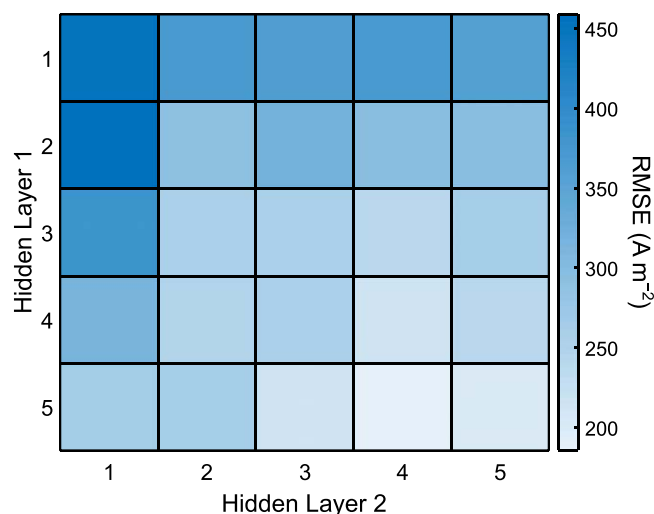| ($n = 50$) RMSE (A m^{-2})	Activation Functions		
	ReLU	Hyperbolic Tangent	Sigmoid
Training	537.3 ± 142.4	214.1 ± 22.8	184.9 ± 16.7
Validation	434.0 ± 47.5	214.1 ± 18.1	185.5 ± 11.7
Testing	610.1 ± 125.8	206.9 ± 23.0	179.1 ± 18.2

Figure 4. Grid representation of RMSE values for the validation data as a function of the number of nodes in the two hidden layers.

configuration of five nodes in the first hidden layer and four nodes in the second hidden layer registered the smallest RMSE for the validation data, and this configuration was used for subsequent analysis of the testing data. Across the rectifier linear unit (ReLU), hyperbolic tangent, and sigmoid activation functions, we found the selection of the sigmoid resulted in the smallest RMSE across all data sets (Table II), motivating its use throughout the rest of this study. The resulting mean validation RMSE is shown in Fig. 4 after 50 iterations of training to account for differences in NN initializations using the sigmoid activation function.

Subsequently, the testing data were passed through the trained NN and compared with the resulting predicted values (Fig. 5). The parity plot of the testing data in Fig. 5a demonstrates that the model has good predictive ability accruing an RMSE of 179.1 ± 18.2 A m^{-2} (17.91 ± 1.82 mA cm^{-2}). In addition to predicted values, the difference between the predicted and actual values can be compared to the model distribution, as shown in Fig. 5b, with the spread of the residuals has a near-Gaussian distribution with large tails (Fig. 5c). The analyses of the parity plot and the residuals indicate that the selected model architecture can describe the underlying relations between the electrode property inputs and *i*.
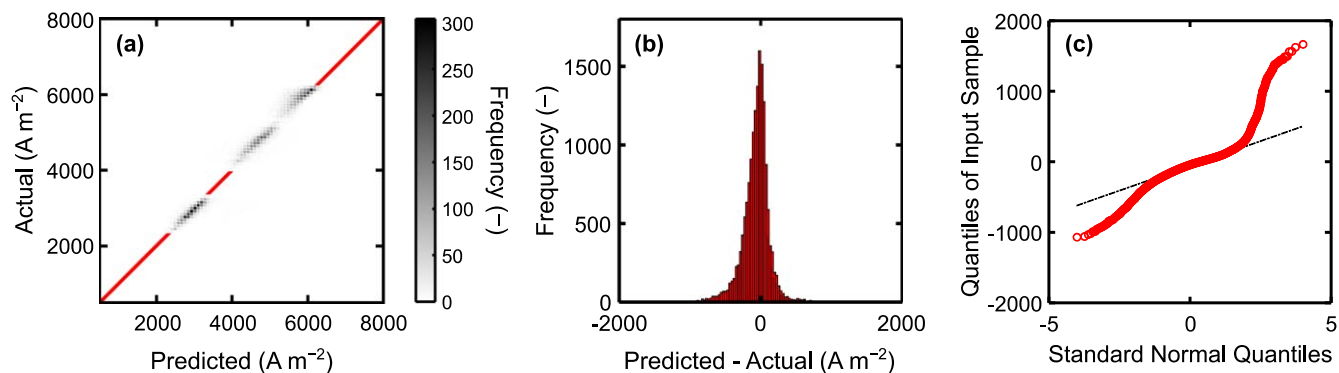

Figure 5. (a) The parity plot of the testing data using a NN with five nodes in the first hidden layer and four nodes in the second hidden layer. As a result, (b) the histogram of the testing data residuals with a near Gaussian-like distribution. The quantile-quantile plot reveals large tails on the distribution of the residuals.
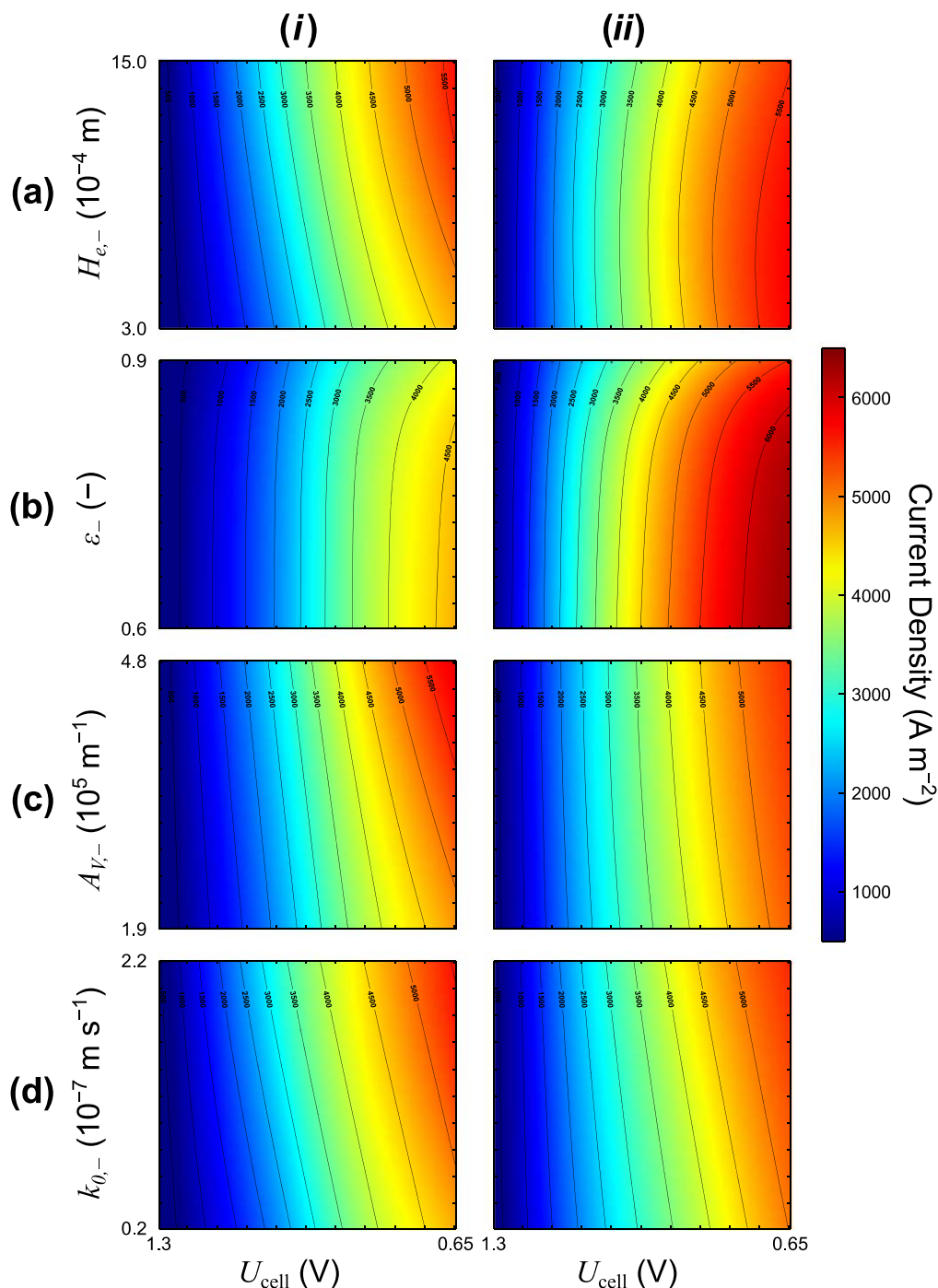
Figure 6. The incrementally swept current responses to changes U_{cell} with respect to (a) $H_{e,-}$, (b) ε_{-}, (c) $A_{V,-}$, and (d) $k_{0,-}$ with the other parameters fixed at their global (i) minimum and (ii) maximum value and the contours representing iso-current-density lines.

Individual parameter sweeps.—Individually, all input parameters are known to have various correlative strengths with respect to i (Table II). However, the degree to which the polarization curve topology changes is obfuscated by the highly nonlinear relationships and complicates optimization procedures. The intrinsic benefit of the NN model is its use for additional analyses without the large computational overhead of high-dimensional models. To demonstrate this, surface plots depicting the polarization with individual parameter sweeps are shown in Fig. 6. The parameters are incrementally stepped from their minimum to maximum values with all other input parameters fixed at either their (i) lowest or (ii)

highest value, as represented by the two columns in Fig. 6. The changes between Figs. 6ai and 6aii reveal the influence of the other parameters on i as $H_{e,-}$ is varied; specifically, the high i region expands in Fig. 6ai, suggesting at low values of ε_{-}, $A_{V,-}$, and $k_{0,-}$, increasing $H_{e,-}$ yields higher values of i. Interestingly, Fig. 6aii reveals that at large values of ε_{-}, $A_{V,-}$, and $k_{0,-}$, the high i region does not continuously increase with greater $H_{e,-}$, suggesting an optimum $H_{e,-}$ that corresponds to enhanced cell performance.

Analogously, the individual parameter sweeps of ε_{-}, $A_{V,-}$, and $k_{0,-}$ are shown in Figs. 6b–6d, respectively. The ε_{-} for high values of the other input parameters (Fig. 6bii) reveals a large high i region

Table IV. Input values used to generate the maximum power densities shown in Fig. 7.

	Units	Training	Validation	Testing	GA	COMSOL
$H_{e,-}$	10^{-4} m	10	6	8	6.62	6.62
ε_-	—	0.60	0.68	0.62	0.60	0.60
$A_{V,-}$	10^5 m^{-1}	4.80	4.60	4.30	4.80	4.80
$k_{0,-}$	10^{-7} m s^{-1}	2.20	1.90	2.10	2.20	2.20
U_{cell}	V	0.90	0.80	0.85	0.81	0.81
i	10^3 A m^{-2}	4.82	5.57	5.23	5.64	5.58
P	10^3 W m^{-2}	4.34	4.46	4.45	4.58	4.53
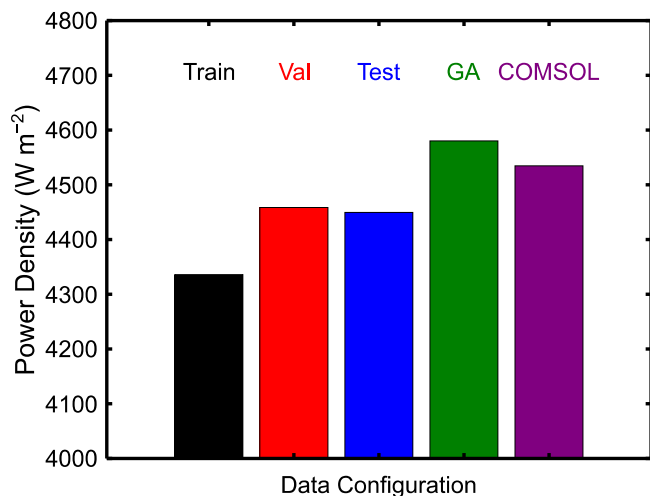
Figure 7. The maximum values of the power density P across the training, validation, and testing data, along with the output GA values. The COMSOL® P value is calculated by inputting the optimal GA values into a previously reported 2D COMSOL® vanadium redox flow cell model.

Table V. The estimated time required to acquire the data and conduct the analyses in this study, assuming identical resources for 2,648 data points generated, as well as 323,208 and ca. 900,000 simulations required for the individual parameter sweeps and GA, respectively.

Time (h)	3D Model	2D Model	This work
Data Generation	1.5×10^3	1.5×10^1	3.7×10^1
NN Training	—	—	1.0×10^0
Single Parameter	1.8×10^5	1.8×10^2	4.4×10^{-3}
GA	5.0×10^5	5.0×10^2	8.3×10^{-2}
Estimated Total	6.8×10^5	6.8×10^2	3.7×10^1

when ε_- is low, which aligns with intuition in that smaller values of ε_- corresponding to elevated i. For both $A_{V,-}$ and $k_{0,-}$, the high i region increases with larger quantities of $A_{V,-}$ and $k_{0,-}$ regardless of the fixed values of $H_{e,-}$ or ε_-, suggesting both $A_{V,-}$ and $k_{0,-}$ would continue to increase to achieve larger regions of high i.

The contours of the individual parameter sweep polarizations depict the degree to which each quantity impacts i. However, the exact value of the parameters held constant can influence the sensitivity of the sweeping parameter. For example, the correlative strengths change across different polarization regimes, particularly when entering the mass-transfer-limiting regime.[27] In such cases, the values of the parameters normally held constant should also be modified to better elucidate how the relative importance of each parameter changes in these distinct regions. The example in this study only contemplates the input parameter extrema, but additional system parameters that invoke transitions into different physical domains (e.g., mass-transfer-limiting, non-Darcy flow, etc.), could also be considered, provided the chosen fitness function is accurate in those domains.

The benefit of this analysis is the granularity in individual parameter and U_{cell} value changes, allowing for finer observations of the i response. However, we note that the efficacy of such analyses is contingent upon the model accuracy and the reliability of the data used in its generation. In concert with subsequent optimization procedures, this protocol can provide an intuitive glance into relative importance of different parameters throughout cell polarization.

Genetic algorithm.—The NN is also a tool with which to perform high-throughput computations otherwise impossible with higher order models. A GA can leverage the dexterity of the NN to comb through the input space and report outputs subject to a defined objective function. Although a GA does not guarantee convergence to a global optimum within a pre-specified number of iterations, it will converge to a global optimum with probability one as its tuning parameters are changed in a known fashion. By reducing the computational cost of the optimization objective by several orders of magnitude, the GA can run many more iterations within the specified time, thus trending towards a global optimum. Provided the fast objective function is a close approximation of the true objective, the GA will converge to a solution close to the global optimum of the original computationally expensive simulation.

For this GA implementation, the objective was to maximize P by altering the NN input values, as we hypothesized the posterior calculation of P would be less nonlinear than computing P within the NN directly. Additionally, the GA protocol was bounded by the minimum and maximum parameter values of the training data set (Table I) to limit the search space. As with the NN training and validation procedure, we employed multiple starts ($n = 100$) from a random seed to account for variations in the GA initializations. We recorded the maximum P across all acquired data and compared this value to those from the training, validation, and testing data sets (Fig. 7), revealing the GA identifies a greater P. Observing the corresponding input values leading to the higher P (Table IV), we found that the $H_{e,-}$ settled on a midpoint value of 6.62×10^{-4} m at a U_{cell} of 0.81 V, with all other values lying on their respective bounds. Indeed, when the values of the GA are input into the 2D COMSOL® model for verification, the resulting P is greater than those in the original data sets, although less than the maximum value of the GA, which we attribute to inaccuracies in the NN. Overall, this approach succeeded in screening the electrode design space subject to an objective to provide a verified improvement throughout the data sets. However, we note that different objective functions relating to fluid dynamics[27] or system cost[3] could be used for additional property optimization protocols.

Using the NN affords substantial computational reductions for higher dimensional multi-physics modeling of the flow cell domain. Cheng et al.[27] report that the times required to compute one point on a polarization curve for the 2D and 3D models are ca. 2×10^1 s and ca. 2×10^3 s, respectively,[27] compared to ca. 2×10^{-2} s with the NN. However, we note that the 2D model was used to generate the

data for the NN training, validation, and testing, which took ca. 37 h to complete in addition to the ca. 1 h required to train the NN and select the architecture. Despite these added time costs, the NN model vastly reduces the computational expense for completing the individual parameter sweeps and the GA. The total computational time to perform these calculations are shown in Table V; though, we note that differences in the computer processors used in the high-dimensional models and the NN generation may influence these values. As such, the COMSOL®-computed times are propagations of the quantities presented in Cheng et al.[27] multiplied by the data sizes in this study. This consideration notwithstanding, the NN model can describe the system nonlinearities at high granularity, which can become cumbersome without access to high-performance machines.

Importantly, while P represents the peak power density for this study, it does not represent the peak power density for all potential electrodes or redox flow cells. Specifically, within this modeling and optimization framework, only the negative electrode is modeled while the positive electrode is held constant. Other system parameters, such as active species concentration, electrolyte flow rates, electrode tortuosity, and operating temperature, which can have a substantial impact on overall cell performance, are not yet considered and could be pursued in follow-up work.

Conclusions

In this study, a 2D COMSOL® model was used to generate training, validation, and testing data sets of different negative electrode property sets by adjusting $H_{e,-}$, ε_-, $A_{V,-}$, $k_{0,-}$, and U_{cell}, which we used to build and validate a NN that rapidly predicts i. Given the computational lightness of the NN, we performed individual parameter sweeps to generate polarization plots depicting the changes in i as a function of electrode properties and across different U_{cell} values. These individual parameter sweeps demonstrate the computational lightness of the NN and identify potential parameter improvements, as evinced by the current density contours. Finally, we employed a bounded GA with the objective to maximize P in our NN model by changing model inputs; this approach revealed improved performance relative to the data sets used in generating the NN, demonstrating that smaller values of $H_{e,-}$ can be leveraged for improved cell power density with other parameters intuitively settling on their bounded values. We note that additional electrode and system parameters can be explored, either individually or in aggregate, for further analysis, potentially affording greater gains in electrochemical performance.

The results from the electrochemical model reveal trends in electrode property importance at various positions on the polarization curve, which can inform electrode design selections for flow cells operating at or near the mass-transfer-limiting regime. Additionally, the fitness function of the GA was set to maximize P; other objective functions can be employed to balance the tradeoffs between fluid dynamic, electrochemical performance, and technoeconomic analysis to obtain a more sophisticated optimization of electrode properties. While the data for this study originates from simulations, it is intended to support—not replace—experimental campaigns to which the numerical process in this study could also be applied.

Acknowledgments

CRediT Author Statement

Kevin M. Tenny: Conceptualization, Methodology, Validation, Formal analysis, Investigation, Data curation, Writing—original draft, Writing—review & editing, Visualization, Project administration. **Richard D. Braatz:** Methodology, Writing—original draft, Writing—review & editing. **Yet-Ming Chiang:** Resources, Funding acquisition. **Fikile R. Brushett:** Resources, Writing—original draft, Writing—review & editing, Funding acquisition.

ORCID

Kevin M. Tenny https://orcid.org/0000-0002-3104-3746
Richard D. Braatz https://orcid.org/0000-0003-4304-3484
Yet-Ming Chiang https://orcid.org/0000-0002-0833-7674
Fikile R. Brushett https://orcid.org/0000-0002-7361-6637

References

1. A. Z. Weber, M. M. Mench, J. P. Meyers, P. N. Ross, J. T. Gostick, and Q. Liu, *J. Appl. Electrochem.*, **41**, 1137 (2011).
2. M. L. Perry and A. Z. Weber, *J. Electrochem. Soc.*, **163**, A5064 (2016).
3. J. D. Milshtein, R. M. Darling, J. Drake, M. L. Perry, and F. R. Brushett, *J. Electrochem. Soc.*, **164**, A3883 (2017).
4. K. E. Rodby, T. J. Carney, Y. Ashraf Gandomi, J. L. Barton, R. M. Darling, and F. R. Brushett, *J. Power Sources*, **460**, 227958 (2020).
5. L. F. Arenas, C. Ponce de León, and F. C. Walsh, *J. Energy Storage*, **11**, 119 (2017).
6. D. S. Aaron, Q. Liu, Z. Tang, G. M. Grim, A. B. Papandrew, A. Turhan, T. A. Zawodzinski, and M. M. Mench, *J. Power Sources*, **206**, 450 (2012).
7. T. Y. Ertugrul, J. T. Clement, Y. A. Gandomi, D. S. Aaron, and M. M. Mench, *J. Power Sources*, **437**, 226920 (2019).
8. K. J. Kim, M.-S. Park, Y.-J. Kim, J. H. Kim, S. X. Dou, and M. Skyllas-Kazacos, *J. Mater. Chem. A*, **3**, 16913 (2015).
9. C.-H. Tian, R. Chein, K.-L. Hsueh, C.-H. Wu, and F.-H. Tsau, *Rare Met.*, **30**, 16 (2011).
10. B. Sun and M. Skyllas-Kazacos, *Electrochim. Acta*, **37**, 2459 (1992).
11. K. V. Greco, A. Forner-Cuenca, A. Mularczyk, J. Eller, and F. R. Brushett, *ACS Appl. Mater. Interfaces*, **10**, 44430 (2018).
12. S. Litster and G. McLean, *J. Power Sources*, **130**, 61 (2004).
13. C. R. Dennison, E. Agar, B. Akuzum, and E. C. Kumbur, *J. Electrochem. Soc.*, **163**, A5163 (2016).
14. I. Mayrhuber, C. R. Dennison, V. Kalra, and E. C. Kumbur, *J. Power Sources*, **260**, 251 (2014).
15. S. Liu, M. Kok, Y. Kim, J. L. Barton, F. R. Brushett, and J. Gostick, *J. Electrochem. Soc.*, **164**, A2038 (2017).
16. Q. Wu, Y. Lv, L. Lin, X. Zhang, Y. Liu, and X. Zhou, *J. Power Sources*, **410–411**, 152 (2019).
17. J. Sun, L. Zeng, H. R. Jiang, C. Y. H. Chao, and T. S. Zhao, *J. Power Sources*, **405**, 106 (2018).
18. J. D. Milshtein, K. M. Tenny, J. L. Barton, J. Drake, R. M. Darling, and F. R. Brushett, *J. Electrochem. Soc.*, **164**, E3265 (2017).
19. A. Gayon Lombardo, B. A. Simon, O. Taiwo, S. J. Neethling, and N. P. Brandon, *J. Energy Storage*, **24**, 100736 (2019).
20. C. Ma, X. Li, L. Lin, L. Chen, M. Wang, and J. Zhou, *J. Energy Storage*, **18**, 16 (2018).
21. B. W. Zhang, Y. Lei, B. F. Bai, and T. S. Zhao, *Int. J. Heat Mass Transfer*, **135**, 460 (2019).
22. X. Ma, H. Zhang, and F. Xing, *Electrochim. Acta*, **58**, 238 (2011).
23. M. R. Gerhardt, A. A. Wong, and M. J. Aziz, *J. Electrochem. Soc.*, **165**, A2625 (2018).
24. M. D. R. Kok, R. Jervis, D. Brett, P. R. Shearing, and J. T. Gostick, *Small*, **14**, 1703616 (2018).
25. G. Qiu, C. R. Dennison, K. W. Knehr, E. C. Kumbur, and Y. Sun, *J. Power Sources*, **219**, 223 (2012).
26. M. D. R. Kok, A. Khalifa, and J. T. Gostick, *J. Electrochem. Soc.*, **163**, A1408 (2016).
27. Z. Cheng, K. M. Tenny, A. Pizzolato, A. Forner-Cuenca, V. Verda, Y.-M. Chiang, F. R. Brushett, and R. Behrou, *Appl. Energy*, **279**, 115530 (2020).
28. J. Bao, V. Murugesan, C. J. Kamp, Y. Shao, L. Yan, and W. Wang, *Adv. Theor. Simul.*, **3**, 1900167 (2020).
29. X. Li, K. Li, Z. Yang, and C. Wong, *Advanced Computational Methods in Energy, Power, Electric Vehicles, and Their Integration, Communications in Computer and Information Science.*, ed. K. Li, Y. Xue, S. Cui, Q. Niu, Z. Yang, and P. Luk (Springer, Berlin) 386 (2017).
30. H.-F. Shen, X.-J. Zhu, M. Shao, and H. Cao, *J. Appl. Math.*, **2013**, 1 (2013).
31. K. A. Severson et al., *Nat. Energy*, **4**, 383 (2019).
32. P. M. Attia et al., *Nature*, **578**, 397 (2020).
33. E. Chemali, P. J. Kollmeyer, M. Preindl, and A. Emadi, *J. Power Sources*, **400**, 242 (2018).
34. M.-F. Ng, J. Zhao, Q. Yan, G. J. Conduit, and Z. W. Seh, *Nat. Mach. Intell.*, **2**, 161 (2020).
35. X. Feng et al., *eTransportation*, **3**, 100051 (2020).

36. C. Hu, G. Jain, P. Zhang, C. Schmidt, P. Gomadam, and T. Gorka, *Appl. Energy*, **129**, 49 (2014).
37. Y. Li, C. Zou, M. Berecibar, E. Nanini-Maury, J. C.-W. Chan, P. van den Bossche, J. Van Mierlo, and N. Omar, *Appl. Energy*, **232**, 197 (2018).
38. R. J. May, H. R. Maier, G. C. Dandy, and T. M. K. G. Fernando, *Environ. Modelling Softw.*, **23**, 1312 (2008).
39. J. H. Holland, *Sci. Am.*, **267**, 66 (1992).
40. S. M. Moosavi, A. Chidambaram, L. Talirz, M. Haranczyk, K. C. Stylianou, and B. Smit, *Nat. Commun.*, **10**, 1 (2019).
41. L. Magnier and F. Haghighat, *Build. Environ.*, **45**, 739 (2010).

42. D. F. Cook, C. T. Ragsdale, and R. L. Major, *Eng. Appl. Artif. Intell.*, **13**, 391 (2000).
43. C. Shen, L. Wang, and Q. Li, *J. Mater. Process. Technol.*, **183**, 412 (2007).
44. L. Wang, *Appl. Math. Comput.*, **170**, 1329 (2005).
45. T. Kerdphol, K. Fuji, Y. Mitani, M. Watanabe, and Y. Qudaih, *Int. J. Electr. Power Energy Syst.*, **81**, 32 (2016).
46. Y. Y. Choi, S. Kim, S. Kim, and J.-I. Choi, *J. Power Sources*, **450**, 227684 (2020).
47. Z. Wei, R. Xiong, T. M. Lim, S. Meng, and M. Skyllas-Kazacos, *J. Power Sources*, **402**, 252 (2018).
48. Z. Lei, W. Zhenpo, H. Xiaosong, and D. G. Dorrell, *IFAC Proceedings Volumes*, **47**, 3899 (2014).