

# PEMNET: A Transfer Learning-based Modeling Approach of High-Temperature Polymer Electrolyte Membrane Electrochemical Systems

Luis A. Briceno-Mena<sup>1</sup>, José A. Romagnoli<sup>1</sup>, and Christopher G. Arges<sup>2,\*</sup>

<sup>1</sup> Cain Department of Chemical Engineering, Louisiana State University, Baton Rouge, Louisiana 70803, United States

<sup>2</sup> Department of Chemical Engineering, The Pennsylvania State University, University Park, PA 16802, United States

\* Corresponding author: [chris.arges@psu.edu](mailto:chris.arges@psu.edu)

## ABSTRACT

Widespread adoption of high-temperature electrochemical systems such as polymer electrolyte membrane fuel cells (HT-PEMFCs) requires models and computational tools for accurate optimization and guiding new materials for enhancing fuel cell performance and durability. While robust and better suited for extrapolation, knowledge-based modeling has limitations as it is time consuming and requires information about the system that is not always available (e.g., material properties and interfacial behavior between different materials). Data-driven modeling on the other hand, is easier to implement, but often necessitates large datasets that could be difficult to obtain. In this contribution, knowledge-based modeling and data-driven modeling are combined by implementing a Few-Shot Learning (FSL) approach. A knowledge-based model originally developed for a HT-PEMFC was used to generate simulated data (887,735 points) and used to pretrain a neural network source model tuned via a genetic algorithm-based AutoML. Then, experimental datasets from HT-PEMFCs with different materials and operating conditions (~50 points each) were used to train 6 target models via FSL. Models for the unseen data reached high accuracies in all cases (rRMSE < 10%).

**Keywords:** Transfer Learning, Few-shot Learning, AutoML, data-driven modeling, high-temperature polymer electrolyte membrane fuel cells.

## INTRODUCTION

Electrochemical systems for energy conversion and storage, as well as manufacturing, powered on solar, nuclear and wind are central to reducing greenhouse gas emissions in addition to attaining a more sustainable way of living. For example, fuel cell electric vehicles using green hydrogen pose a great opportunity to curtail greenhouse gas emissions that hail from transporting goods via heavy duty vehicles (e.g., trucking), marine ships, aviation, and trains<sup>1-3</sup>. In recent years, high-temperature polymer electrolyte membrane (HT-PEM) fuel cells have experienced a renewed interest since the advent of ion-pair PEM architectures that expand the temperature range for these platforms and tolerance to higher levels of humidity<sup>4-7</sup>. Pairing HT-PEMs with phosphonated ionomer electrode binders in the past year have resulted in high performing fuel cells. For example, a peak power of 1.7 W cm<sup>-2</sup> has been achieved for HT-PEMFCs using hydrogen and oxygen<sup>5</sup>.

A central advantage of HT-PEM fuel cells is their ability to tolerate carbon monoxide (CO) in the hydrogen stream as CO adsorption at temperatures of 200 °C or above is diminished<sup>8</sup>. Over 90% of all hydrogen is derived from steam methane reforming (SMR) that leads to mixtures of hydrogen and CO. Electrochemical process that tolerate CO allow the use of low-cost hydrogen because hydrogen from SMR is about 2x to 3x cheaper than hydrogen from water electrolysis. Furthermore, an added benefit of higher temperature operation for fuel cell electric vehicle stacks is that it simplifies heat management as the larger temperature gradient favors greater heat rejection and the ion-pair HT-PEMs do not require humidification for ionic conduction. This latter attribute eliminates an external humidifier, an ancillary unit that adds cost. Removal of ancillary units also simplifies the balance of plant for the fuel cell system. Despite these extraordinary developments in a relatively short-period of time, further maturation of HT-PEMFC platforms is needed to reduce platinum group metal (PGM) loading to reduce overall system capitals costs while maintaining or exceeding the demonstrated performance.

Lowering PGM loadings requires greater electrocatalyst utilization in the electrode layers of HT-PEMFCs while also curtaining interfacial resistances in the electrode layers related to reactant, electron, and ion transport. New materials, such as ionomer electrode binders, offer tremendous opportunity to enhance electrocatalyst utilization while co-currently addressing interfacial

resistances; however, the material properties and how they affect interfacial reaction kinetics and transport behavior is poorly understood under at various cell operating conditions (i.e., temperature, pressure, stoichiometric ratio, etc.). This poor understanding typically leads to an ‘Edisonian’ approach of testing new materials and observing performance with little attention given to modeling the systems and trying to bridge the gap between materials properties and device level performance. Additionally, the approach of synthesize and test gives limited knowledge in short period of time and stymies the development and optimization of system performance to specified constraints (e.g., PGM loadings). Accelerating the maturation of HT-PEM fuel cells with new materials, such as phosphonated ionomer electrode binders and ion-pair membranes, requires a comprehensive and computationally inexpensive model usable for optimization but also capable of capturing the properties of new materials and their influence on cell performance.

Since a physical understanding of the system is required for this approach, the modeling task is complex and time consuming, and sometimes not all necessary information is available for capturing all the relevant interactions among the variables being modeled<sup>9</sup>. Furthermore, for knowledge-based models, new experimental findings relevant to a specific component of the system cannot be easily incorporated into an existing model, and data on the impact of these on the overall system must be collected as well. Conversely, the data-driven approaches, which includes Machine Learning (ML), are used to obtain a model from the raw experimental data<sup>10</sup>. This approach has the advantage that given enough data, the model could potentially represent all the interactions among the system’s variables within the range of the data. However, the amount and variety of data required for the actual implementation of this approach in nascent HT-PEM electrochemical devices can be cost-prohibitive. This latter issue, also called scarce availability of data, is prevalent in many engineering applications and prevents the full exploitation of data-driven modeling. For example, full scale operation data for the batch production of new fine chemicals and materials is often scarce because generating such data can be costly. This problem can be addressed by exploiting available operation data from similar products resulting in shorter timelines to improved product development and scale-up of the entity of interest. Another example is that of adaptive model predictive control as processes change over time for many reasons, and data-driven models used for process control will not have enough data to accurately represent the new states of the plant<sup>11</sup>. Recently, we reported a framework that takes advantage of the benefits

of both knowledge-based and data-driven approaches by hierarchically combining data-driven modeling for materials behavior and knowledge-based modeling for fuel cell device performance<sup>12</sup>. While this approach helped reduce the time and the amount of experimental data required for the model development, some aspects like the incorporation of new knowledge and the flexibility of the modeling approach needs further improvement. In this work, we demonstrate that Transfer Learning can be used to address these shortcomings.

Transfer Learning (TL)<sup>13</sup> is a ML technique derived from the notion of transfer of learning first proposed in the educational research context<sup>14</sup>. In TL, a data-driven model previously trained (general training) for a given task (source domain) is used as the base to build a model for a new task (target domain), with less data being required for the new training stage (task-specific training). Typically, the source and target domains are similar or closely related. As an illustrative example, a large database of labeled images of dogs (source domain) could be used for pretraining a learning agent to classify breeds, and then repurposed to classify images of cats (target domain). Formally, given a source domain  $\mathcal{D}_s$ , and learning task  $\mathcal{T}_s$ , a target domain  $\mathcal{D}_T$  and a learning task  $\mathcal{T}_T$ , transfer learning aims to help the learning of the target predictive function  $f_T(\cdot)$  for the target domain using the knowledge in  $\mathcal{D}_s$  and  $\mathcal{T}_s$ , where  $\mathcal{D}_s \neq \mathcal{D}_T$  and  $\mathcal{T}_s \neq \mathcal{T}_T$ <sup>15</sup>. In this work, the source domain  $\mathcal{D}_s$  and learning task  $\mathcal{T}_s$  come from simulations generated using a low fidelity knowledge-based model. The target domain  $\mathcal{D}_T$  and learning task  $\mathcal{T}_T$  come from experimental data. A useful extension of TL is the so-called few-shot learning (FSL) in which the task-specific training stage uses a very small amount of data (i.e., on the order of  $1 \times 10^1$ )<sup>15</sup>. By using FSL, new findings can be easily incorporated with little experimental cost. Furthermore, the information contained in a model for a given device can be readily transferred to a model for different, yet similar, device architecture<sup>16, 17</sup>.

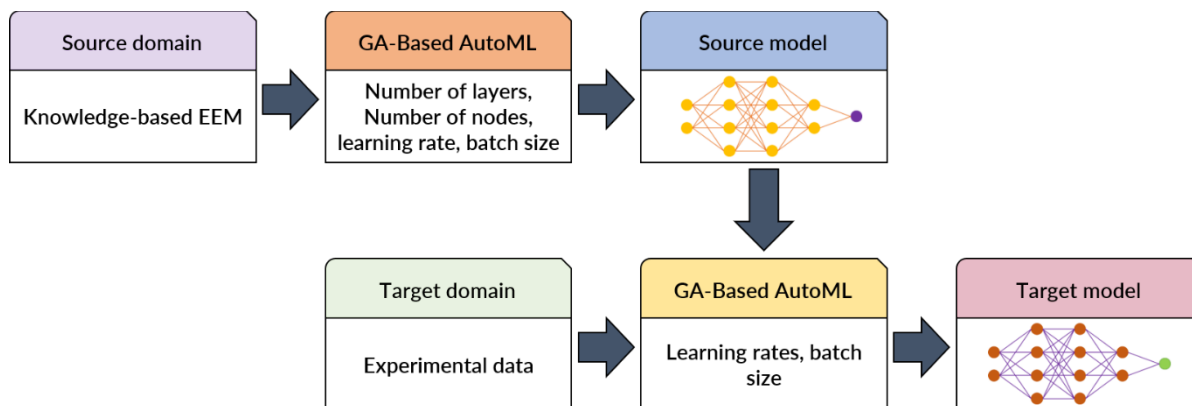
Although the task-specific training can be carried out with a small data set from the target domain, the general training still requires large amounts of data from the source domain. In some applications like image or speech recognition, an existing large dataset can be used to pretrain the model and then proceed with the transfer<sup>18, 19</sup>. However, extensive data sets for HT-PEM-based electrochemical systems are not available and must be generated. Using an existing knowledge-based model, large and balanced datasets can be produced to support the pretraining stage.

Furthermore, the datasets come from knowledge-based simulations, and they generate useful knowledge for new model developments.

In this contribution, we present PEMNET, a TL approach for the modeling of HT-PEMFCs that uses a knowledge-based model validated for our previously reported HT-PEMFC data as source domain. First, a strategy to leverage a knowledge-based model to generate a useful dataset is discussed. Second, since tuning the hyperparameters of the ML architectures is not a trivial task, a genetic algorithm-based optimization implementation for AutoML is presented. Finally, the applicability of PEMNET to model fuel cell devices is demonstrated by obtaining models for 6 HT-PEMFCs with different membrane and ionomer binder chemistries.

## METHODS

The overarching strategy of this implementation is described in **Figure 1**. A knowledge-based explicit equations model (EEM) for a HT-PEMFC was used as the source domain. Several datasets for different membrane electrode assemblies (MEAs) and temperatures were used as target domains for a total of 6 new models produced. A fully connected neural network optimized using a genetic algorithm-based AutoML was implemented.

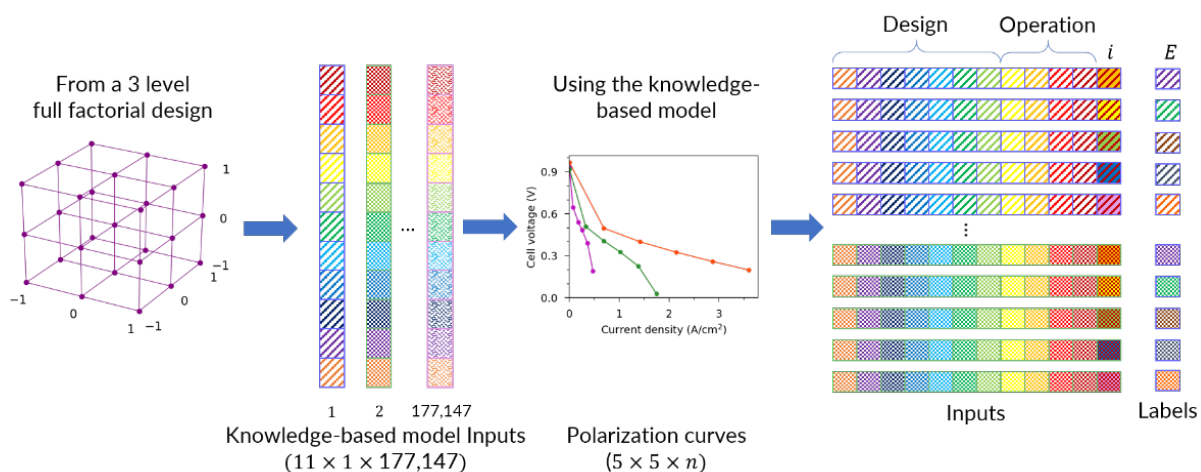


**Figure 1.** Transfer learning-based modeling strategy.

## Dataset Preparation

### HT-PEMFC Simulated Data

For the general training stage, simulated data was generated using the knowledge-based model reported elsewhere<sup>12</sup>. **Figure 2** shows the general procedure for data preprocessing. To obtain a balanced dataset, the values for the input variables were structured following a full 3-level factorial experimental design using 11 variables ( $3^{11}$ ) (see **Table 1**). The full factorial design is considered sufficient to ensure a uniform distribution of the data and enough examples for training so that pretraining data is not a limiting factor for the subsequent steps. The levels for the factorial design were determined based on authors' experience and considering feasible ranges for each variable. It is important to note that all HT-PEMFC models and data only consider pure oxygen as the oxidant. For each set of inputs, 5 points in the polarization curves were generated. The resulting datasets were arranged to generate a total of 887,735 input vectors of size 12 and the corresponding labels of size 1. Data were scaled using over min-max scaling in all cases.



**Figure 2.** Simulated data generation and preprocessing strategy. The values for the inputs are structured following a  $3^{11}$  factorial design of experiments and fed into the knowledge-based model to generate 5 points in the corresponding polarization curves. The final dataset has  $177,147 \times 5 = 887,735$  samples with input vectors of size 12 and labels of size 1.

**Table 1.** Variables and levels for the fully 3-level factorial experimental design used to structure the simulated dataset.

| Variable                                 | Levels          | Units         |
|--|-----------------|---------------|
| Hydrogen stoichiometric ratio, $S_{H_2}$ | [1, 1.5, 2]     | Dimensionless |
| Oxygen stoichiometric ratio, $S_{O_2}$   | [2, 2.5, 3]     | Dimensionless |
| Temperature, $T$                         | [423, 463, 503] | K             |
| Pressure, $P$                            | [1, 1.5, 2]     | atm           |

|  |                                      |                |
|--|--------------------------------------|----------------|
| Membrane ion exchange capacity, $IEC_{mem}$                | [1.5, 2.25, 3]                       | mequiv/g       |
| Ionomer binder ion exchange capacity, $IEC_{io}$           | [1.5, 2.25, 3]                       | mequiv/g       |
| Membrane thickness, $\delta_{mem}$                         | [0.001, 0.005, 0.01]                 | cm             |
| Ionomer binder thickness, $\delta_{io}$                    | $[5 \times 10^{-7}, 0.0001, 0.0002]$ | cm             |
| Carbon monoxide to hydrogen ratio in fuel stream, $CO/H_2$ | [0, 0.05, 0.1]                       | Dimensionless  |
| Anode catalyst loading, $L_{can}$                          | [0.1, 0.35, 0.6]                     | $mg_{Pt}/cm^2$ |
| Cathode catalyst loading, $L_{cat}$                        | [0.1, 0.35, 0.6]                     | $mg_{Pt}/cm^2$ |

### *Experimental data*

**Table 2** summarizes the experimental data sources and its characteristics. For the HT-PEMFC models, data reported by Atanasov et al.<sup>5</sup> and Venugopalan et al.<sup>6</sup> were used. **Table 2** also notes the binder and HT-PEM chemistry differences. All electrocatalysts were platinum/platinum group metal (PGM) alloy nanoparticles decorated on high-surface area graphitic carbon supports.

### *Training, testing and validation datasets*

Both experimental and simulated data were divided into training and testing sets. For training, a 70:30 ratio was used for training and testing respectively. A 5-fold cross validation was performed on all models and used to report model performance as the relative root mean squared error (rRMSE). Additionally, a validation set, which included a complete polarization curve at a given temperature, was left out. The purpose of the validation set was to evaluate the performance of the model predicting the polarization at a temperature that was not included in training, thus providing more information about the models' ability to generalize and accurately predict data it has not seen.

**Table 2.** Experimental datasets for few-shot learning.

| ID        | $S_{H_2}$ | $S_{O_2}$ | $T$ (°C)           | $P$ (atm) | HT-PEM <sup>a</sup> | $IEC_{mem}^b$ (mequiv/g) | Electrode ionomer binder <sup>c</sup> | $IEC_{io}^d$ (mequiv/g) | $\delta_{mem}$ (cm) | $\delta_{io}$ (cm) | $L_{can}$ (mg <sub>PtRu</sub> /cm <sup>2</sup> ) | $L_{cat}$ (mg <sub>PtRu</sub> /cm <sup>2</sup> ) | Electrode ionomer binder loading (wt%) |
|-----------|-----------|-----------|--------------------|-----------|---------------------|--------------------------|---------------------------------------|-------------------------|---------------------|--------------------|--|--|--|
| $MEA_0^6$ | 1.2       | 2.2       | 160, 200, 220      | 1.59      | PA-QPPSf-PBI        | 7.9                      | PA-QPPSf                              | 8.9                     | 0.005               | 0.0001             | 0.5<br>37% Pt/C                                  | 0.5<br>37% Pt/C                                  | 30                                     |
| $MEA_1^5$ | 1         | 1         | 120, 160, 200      | 1.47      | PA-PBI              | 9.1                      | PTFE                                  | n/a <sup>e</sup>        | 0.005               | 0.0001             | 1.0<br>Pt/C                                      | 0.75<br>Pt-alloy                                 | n/a <sup>e</sup>                       |
| $MEA_2^5$ | 1         | 1         | 120, 160, 200      | 1.47      | PA-PBI              | 9.1                      | PTFSPA                                | 2.2                     | 0.005               | 0.0001             | 0.5<br>50%PtRu/C                                 | 0.6<br>60% Pt/C                                  | 10.4                                   |
| $MEA_3^5$ | 1         | 1         | 120, 160, 200, 240 | 1.47      | PA-QAPOH            | 7.1                      | PA-QASOH                              | 1.9                     | 0.004               | 0.0001             | 0.5<br>50%PtRu/C                                 | 0.6<br>60% Pt/C                                  | 10.4                                   |
| $MEA_4^5$ | 1         | 1         | 120, 160, 200, 240 | 1.47      | PA-QAPOH            | 7.1                      | PTFSPA                                | 2.2                     | 0.004               | 0.0001             | 0.5<br>50%PtRu/C                                 | 0.6<br>60% Pt/C                                  | 10.4                                   |
| $MEA_5^5$ | 1         | 1         | 120, 160, 200, 240 | 1.47      | PA-TPP/Nafion       | n/a <sup>e</sup>         | PA-QASOH                              | 1.9                     | 0.008               | 0.0001             | 0.5<br>50%PtRu/C                                 | 0.6<br>60% Pt/C                                  | 10.4                                   |

<sup>a</sup> HT-PEM chemistries - PA-PBI: Phosphoric acid (PA) imbibed polybenzimidazole (PBI), PA-QPPSf-PBI: PA imbibed quaternary benzyl pyridinium Udel<sup>®</sup> poly(arylene ether sulfone) (QPPSf)-PBI, PA-QAPOH: PA imbibed quaternary alkyl ammonium poly(phenylene), PA-TPP/Nafion: PA imbibed tin pyrophosphate-Nafion<sup>™</sup> composite,

<sup>b</sup>  $IEC_{mem}$  is based upon the number of H<sub>3</sub>PO<sub>4</sub> mequiv per weight of H<sub>3</sub>PO<sub>4</sub> imbibed HT-PEM otherwise noted

<sup>c</sup> Ionomer binder chemistries - PTFSPA: poly(tetrafluorostyrene phosphonic acid-*co*-pentafluorostyrene), PA-QASOH: PA imbibed quaternary benzyl ammonium polystyrene, PA-QPPSf: PA imbibed QPPSf

<sup>d</sup>  $IEC_{io}$  is based upon H<sub>3</sub>PO<sub>4</sub> or phosphonic acid mequiv per weight of polymer (including acid imbibing if applicable)

<sup>e</sup> n/a – not applicable as the value is either zero or unknown.

## Artificial Neural Networks Architecture

Artificial neural networks (ANN) are a ML method in which a model is built in the form of nodes connected by edges and organized in layers (also called fully connected layers)<sup>20</sup>. In each node, the outputs from the nodes in the previous layers are multiplied by its corresponding weights, which are represented by the edges, added, and then transformed by an activation function to feed the next layer of nodes. ANN are trained by calculating the predicted output for a given set of inputs, computing the error  $E$  between the result and the true value (label), and adjusting the weights accordingly. The weights between nodes  $i$  and  $j$ ,  $w_{ij}$ , are updated using the gradient descent algorithm:

$$w_{ij}^{updated} = w_{ij} + -lr \frac{\partial E}{\partial w_{ij}} \quad (1)$$

Here,  $lr$  is the learning rate and  $\frac{\partial E}{\partial w_{ij}}$  is the partial derivative of the error between the predicted value and the label with respect to each weight. **Eq. (1)** conveys that a greater learning rate leads to a larger change in the weights after each iteration. An extension of the gradient descent algorithm is the so-called mini-batch gradient descent which reduces the variance in the estimate of the gradient by processing a subset of instances (batches) at each iteration<sup>21</sup>. The mini-batch gradient descent equation is shown below:

$$w_{ij}^{updated} = w_{ij} + lr \left(\frac{1}{n}\right) \sum_{m=n \times k}^{(k+1) \times n} \left(\frac{\partial E}{\partial w_{ij}}\right)_m \quad (2)$$

Where  $n$  is the batch size,  $k = \left\lceil 1, \frac{N}{n} \right\rceil$  is the number of batches for a dataset of size  $N$ . In this work, the mini-batch gradient descent algorithm was used via the Adam optimizer<sup>22</sup>. An important aspect of the use of Adam is the effect of batch size during training as it has an influence on both the convergence of the training and the generalizability of the resulting model.

### *Model hyperparameters and AutoML*

A challenge in the implementation of ANN is the tuning of the model hyperparameters, that is, the number of hidden layers ( $L$ ), number of nodes ( $N$ ), the learning rate ( $lr$ ), and the batch size ( $n$ ). Additional elements of the model such as the activation function need also to be specified. Since

the best combination of hyperparameter will be different for each application, there are not too many guidelines as to what could work best. Because of this, obtaining the appropriate set of hyperparameters for a given application can become time consuming. Although random search and grid search is widely used to explore combinations of hyperparameters, evolutionary algorithms have been shown to perform better for multi-objective hyperparameter optimization<sup>23</sup>. Here, the selection of  $L$ ,  $N$ ,  $lr$  and  $n$  is defined as a Mixed Integer Nonlinear Programming problem and solved using a multi-objective evolutionary algorithm, namely the Non-dominated Sorting Genetic Algorithm (NSGA-II)<sup>24</sup>, as implemented by Pymoo<sup>25</sup>. For a multi-objective optimization problem, a solution  $x^1$  is said to dominate another solution  $x^2$  if (i)  $x^1$  is no worse than  $x^2$  for all objectives and (ii)  $x^1$  is better than  $x^2$  in at least one objective. NSGA-II exploits this idea to rank the solutions according to their level of non-domination. Additionally, a crowding distance value (the Manhattan distance in the objective space) is also use for sorting and selecting the best solutions, serving as an explicit diversity preservation mechanism. An important characteristic of NSGA-II is that elites of a population are allowed to pass to the next generation, which can prevent the loss of good solutions. A condensed description of NSGA-II is provided in **Supplemental Information**. For tuning the hyperparameters of the neural network, the optimization problem is formulated as follows:

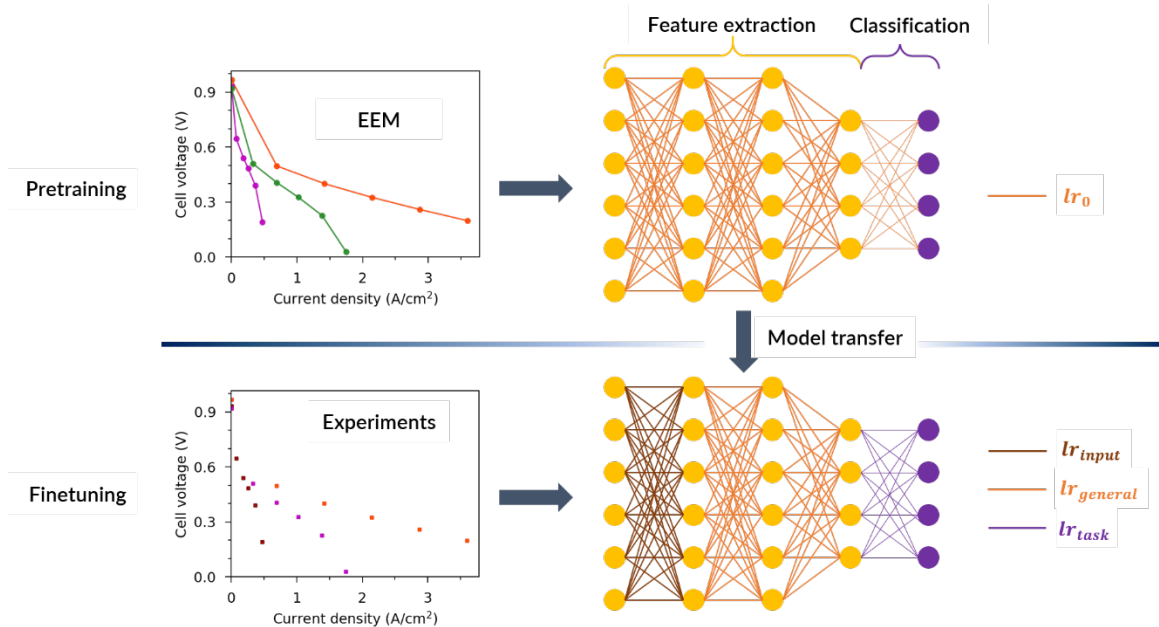
$$\text{Optimization problem 1} \quad \left\{ \begin{array}{ll} \min_x & f_m(x), \\ \text{Subject to} & x \in R^4, \\ & x_i \in \mathbb{Z}, i = 1, 2, 3 \end{array} \right.$$

Where,  $x_1$ ,  $x_2$ ,  $x_3$  are  $L$ ,  $N$  and  $n$  respectively and  $x_4$  is the learning rate.  $f_1(x) = E_{training}^s$ ,  $f_2(x) = E_{testing}^s$ , and  $f_3(x) = E_{validation}^s$  are the mean values for training, testing and validation errors in the 5-fold cross validation. The testing and validation errors are included in the multi-objective optimization problem to prevent overfitting. To obtain a model for a given source domain, a new neural network is built using the set of hyperparameters  $x$  (in addition to a fixed input layer 12:  $N$  and 2 fixed output layers  $N: \text{floor} \left[ \frac{N}{2} \right] : 1$ ) and then its performance is cross validated.

### *Few-Shot Learning*

FSL strategies are broadly based upon the notion of preserving some of the information obtained through a first stage of training (i.e., a source model) and reusing it in some fashion to develop a

target model. In this work, FSL was implemented through regularization. In the regularization approach, the target model has the same architecture as the source model (**Figure 3**), and the learned parameters (weights) are adjusted during a new training stage but some restriction is applied to prevent the model from overfitting the new data<sup>15</sup>. In ANN, one form of regularization involves freezing most of the weights of the model and allowing only the weights of the upper layers of the model to be updated using the new data. While this approach works in some cases, figuring out which layers to freeze can be difficult. Another form of regularization, called differential learning rate, updates all weights during the task-specific training stage, but at a different rate (**Figure 3**), which helps control the influence of the new data over the model.



**Figure 3.** Schematic representation of the transfer learning approach for model development. Top row: pretraining using simulated data and a uniform learning ( $lr_0$ ). Bottom row: finetuning using experimental data, a differential learning rate considering 3 sections ( $[lr_{input}, lr_{general}, lr_{task}]$ ), and the addition of layers (dashed box).

As with the hyperparameters tuning, finding the right combination of learning rates for the target training can be a difficult task. Hence, an optimization problem can be formulated to obtain the appropriate learning rates ( $lr_{input}, lr_{general}, lr_{task}$ ) and batch size ( $n$ ) for a given target domain. The target training optimization problem is formulated as follows:

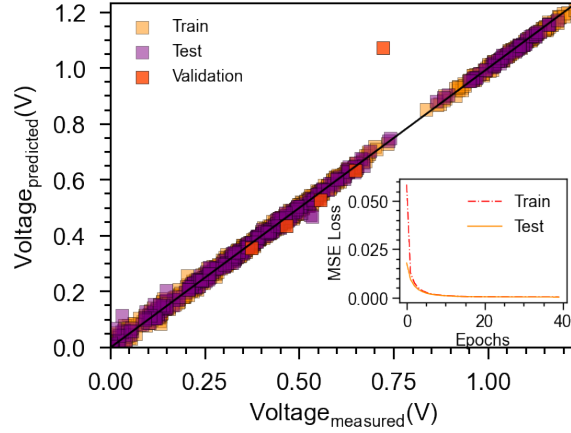
$$\text{Optimization problem 2} \quad \left\{ \begin{array}{l} \min_x \quad f_m(x), \\ \text{Subject to} \quad x \in R^4, \\ \quad \quad \quad x_i \in \mathbb{Z}, i = 4 \end{array} \right.$$

Where,  $x_1, x_2, x_3$  are  $lr_{input}, lr_{general}$  and  $lr_{task}$  respectively and  $x_4$  is the batch size. Same as source training,  $f_1(x) = E_{training}^t, f_2(x) = E_{testing}^t, f_3(x) = E_{validation}^t$  are the mean values for training, testing and validation errors in the 5-fold cross validation.

## RESULTS AND DISCUSSION

### Source model training

The first step in the TL methodology here proposed is to pretrain a neural network using simulated data. From **Optimization Problem 1**, it was found that a neural network with 3 hidden layers, each with 36 nodes (12:36:36:36:18:1), a batch size  $n = 331$ , and a learning rate  $lr = 9.94 \times 10^{-5}$  was the best architecture. For the 5-fold cross validation source model training using the optimized neural network the rRMSE values for training, testing and validation were 2.80%, 2.87% and 29.16% respectively. The simulated data was obtained from an explicit equations model first developed for MEA0 and reported elsewhere<sup>12</sup>. **Figure 4** shows the correspondence between the models' predictions and the training, testing and validation experimental data for the range of voltage. The high error observed at 0.7 and 0.8 V for the validation data, which is the main contributor to the high validation error, stems from the inaccuracy in the original explicit equations model used to generate the simulations for training. As it will be shown in the next section, the validation error drops dramatically after retraining the neural network model using TL and experimental data. **Figure 4** also shows the evolution of the loss curve over the epochs. The loss function, which is reported as the mean squared error loss, decays smoothly, and the train and testing losses reach similar values which serves as an indication that the models do not overfit or underfit the training data.



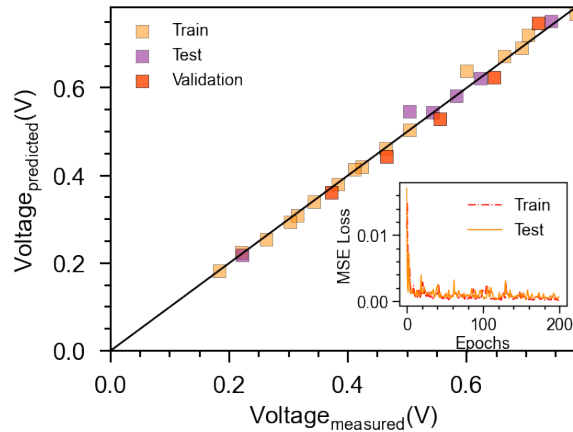
**Figure 4.** Training results for the source model using simulated data from a knowledge-based model and the optimized network architectures. Validation data were MEA0 at 200 °C. Nested plot shows convergence of the loss (MSE Loss) curve.

## Target model training

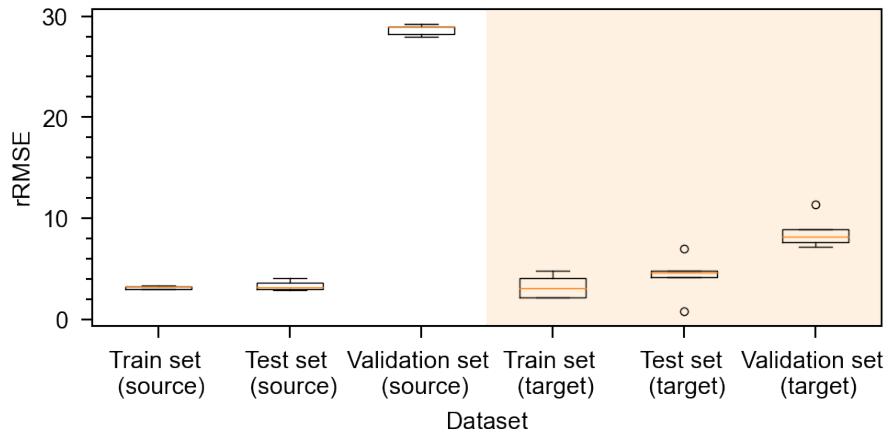
### *Model improvement*

In FSL for finetuning, the target domain is closely related to the source domain. Hereafter the EEM for MEA0 used for pretraining is considered as the source domain, and the experimental datasets (MEA0, MEA1, MEA2, MEA3, MEA4, MEA5) correspond to the target domains. When the source domain is a simulation and the target domain is real experimental data for the same system, TL can be used to obtain a better model. As shown in **Figure 4**, although the optimized neural network reached very low errors for the training and testing data, the error for the validation data is still high. This is consistent with the fact that at the pretraining stage the neural networks have not been exposed to the real experimental data and the source domain is known to have inferior performance at very low current densities. **Figure 5** shows the correspondence between the experimental and the predicted values after target training.

**Figure 6** shows the results for model improvement using the mean rRMSE for the training, testing and validation sets in the 5-fold cross validation as a metric. Here, the optimized learning rates and batch size from **Optimization Problem 2** ( $lr_{input} = 1.99 \times 10^{-8}$ ,  $lr_{general} = 8.80 \times 10^{-6}$ ,  $lr_{task} = 0.00097$ ,  $n = 7$ ) were used. Validation data corresponds to the 200 °C polarization curve. As it can be observed, the validation error decreases by a factor of 3 after target training. These results demonstrate the applicability of TL to improve existing models using small datasets, thus reducing the burden of generating experimental data.



**Figure 5.** Training results for the target model. Validation data were MEA0 at 200 °C. Nested plot shows convergence of the loss (MSE Loss) curve.

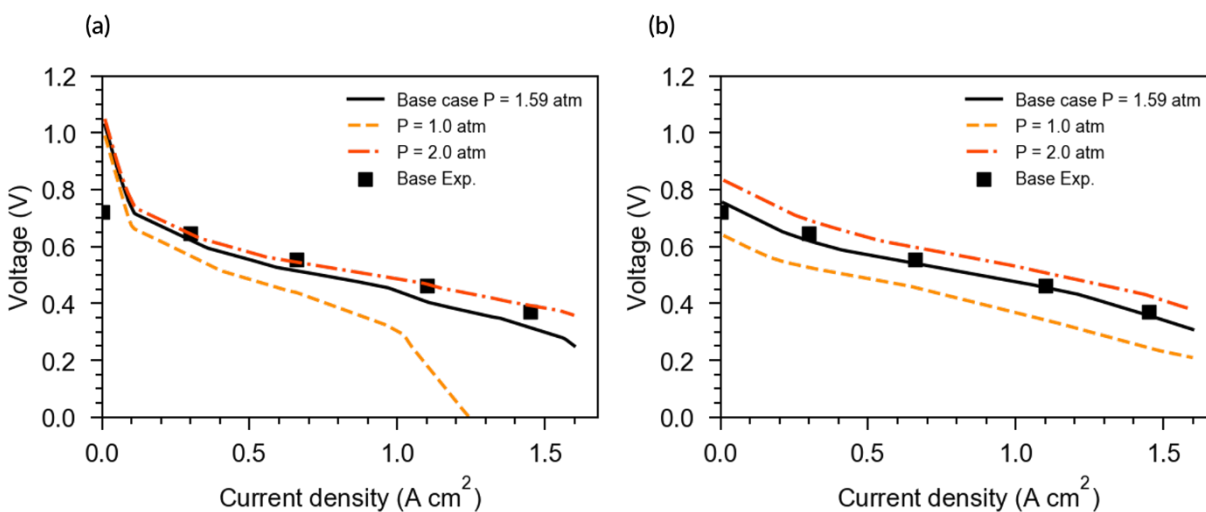


**Figure 6.** 5-fold cross validation results for model improvement. Boxes represent the standard deviation of the errors. Shaded area corresponds to the finetuned model.

Although some of the aspects to the EEM include knowledge-based relationships, some of the components of the EEM are estimated from empirical correlations (e.g., Henry’s constant as a function of acid mass fraction and temperature, limiting current as a function of operating conditions and membrane thickness) and even Machine Learning models such as the prediction of ionic conductivity from ionic exchange capacity and temperature using support vector regression. The ML-based surrogate model obtained via Transfer Learning takes advantage of the knowledge-based information in the EEM while at the same time having more expressivity than the individual semi-empirical correlations, which in turn results in a better representation of the experimental data. Since both the pretraining and target training optimization problems are defined to minimize

training, testing and validation errors, the expressivity of the model can be exploited without overfitting.

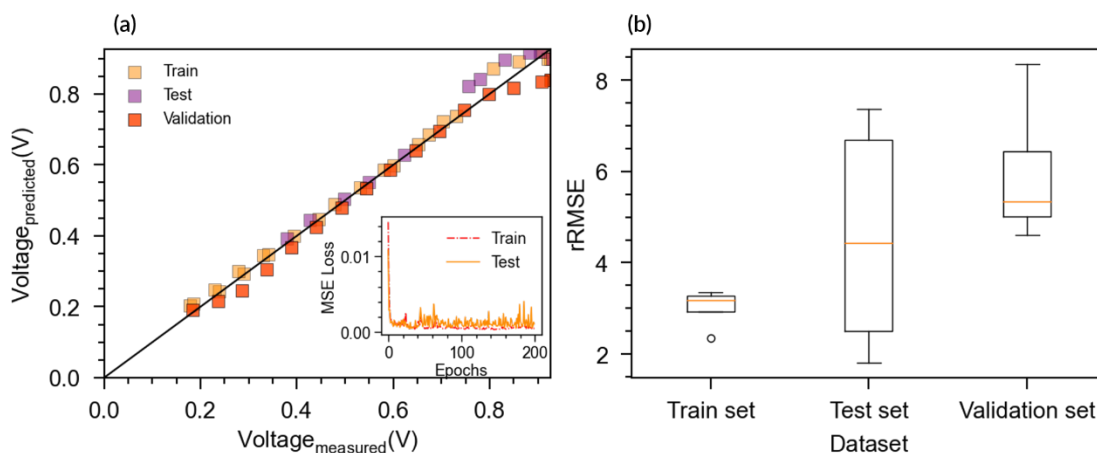
As the EEM models contains physical knowledge (e.g., Butler-Volmer kinetics for informing the activation overpotential and Ohm's Law for determining the ohmic overpotential from electrolyte conductivity)<sup>12</sup>, TL enables the introduction of this information into ML models. An illustrative example of this is shown in **Figure 7**. Here, the change in the polarization curve with changes in pressure is represented for both the source model (left) and the target model (right). Although the target domain does not contain information about the effect of pressure over the system (all examples correspond to the same pressure), the target model preserves some of this knowledge from the pretraining stage, which was introduced via the Nernst potential and physics-informed expressions for the exchange and limiting current density values. Without TL, representing the effect of pressure would require additional experimental data. Additionally, this opens the opportunity for a faster development of models for new materials and device designs and optimal operating parameters, helping guide new materials by identifying the necessary properties for enhancing fuel cell performance (e.g., peak power density or power density at 0.7 V). Furthermore, the TL with ML models can assist with future optimization activities such as pathways to reduce PGM loadings in the MEA while minimizing device performance losses. **Snippet 1 in Supplemental Information** shows an illustrative example of the syntax for source and target model building.



**Figure 7.** Effect of pressure as predicted by the source model (a) and the target model (b). The base case corresponds to the MEA0 at 200 °C.

### Modeling for new materials

TL can be further used to generate new models for different systems. In several of the data sets in **Table 2**, the materials of the MEAs are not the same (e.g., electrode binder, electrocatalyst loading and HT-PEM type); and as such, the different MEA compositions have a profound impact on single-cell polarization as well as power density. To exploit the recent availability of new HT-PEMFC data with ion-pair HT-PEMs and other known HT-PEMs, as well as phosphonated ionomer electrode binders and acid imbibed ionomer electrode binders, the source model was transferred to a target model which was in turn trained with the specific new dataset. **Figure 8** shows the results for the target model training using data for MEA1. Overall, the target model provides a good representation of the fuel cell polarization for the new MEA. The large variability of the cross-validation results in comparison to those of MEA0 show the larger differences between the simulated data (a model created for MEA0) and the new experimental data. Similar results (see **Supplemental Information, Figures S1 to S4**) were obtained for the other MEAs in **Table 2**, showing PEMNET's ability to generate accurate target models for HT-PEMFCs with different materials from a given source model. It is worth noting that a new model is needed only if the nature of the materials change, but not for changes in any of the 11 input variables. For example, if the catalysts is changed (e.g., from Pt to Pt-Ru) a new model (and the corresponding target training) is needed, but if the change is in the amount of catalysts loaded, then there is no need for a new model.



**Figure 8.** Target model obtained using experimental data for MEA1. (a) Model predictions correspondence with experimental data. (b) 5-folds cross validation. Validation data were at 160 °C. Hyperparameters were: ( $lr_{input} = 7.10 \times 10^{-8}$ ,  $lr_{general} = 8.37 \times 10^{-6}$ ,  $lr_{task} = 0.000107$ ,  $n = 4$ ).

## CONCLUSIONS

PEMNET, a Transfer Learning-based approach for modeling HT-PEM electrochemical systems was introduced and its applicability was explored for HT-PEMFCs with different polymer electrolyte MEA chemistries and various cell operating conditions. The implementation uses a knowledge-based model to generate data and pretrain a source model. To obtain a target model, a differential learning rate FSL implementation was used. This enables the faster development of models and has implications for improving HT-PEM fuel cell materials and identifying optimal operating parameters. The ML model architectures and hyperparameters were determined using a genetic algorithm-based AutoML implementation.

A study on a measure of transferability in Transfer Learning modeling is recommended for future investigation.

## ACKNOWLEDGMENTS

This material is based upon work supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under the Advanced Manufacturing Office (AMO) Award Number DE-EE0009101. The views expressed herein do not necessarily represent the views of the U.S. Department of Energy or the United States Government. Luis A. Briceno-Mena thanks the support received from Universidad de Costa Rica.

## SUPPORTING INFORMATION

NSGA-II algorithm, code snippet of PemNet syntax, target training results for MEAs reported in Ref<sup>5</sup>.

## AUTHOR CONTRIBUTIONS

Conceptualization, L.A.B., C.G.A. and J.A.R.; Methodology, L.A.B., C.G.A. and J.A.R.; Investigation, L.A.B. and C.G.A.; Data Curation, L.A.B., C.G.A. and J.A.R.; Visualization, L.A.B., and J.A.R.; Writing – Original Draft, L.A.B., C.G.A. and J.A.R.; Writing – Review & Editing, L.A.B., C.G.A., and J.A.R.; Funding Acquisition, C.G.A., and J.A.R.; Resources, C.G.A., and J.R.; Supervision, C.G.A and J.A.R.

## COMPETING INTEREST STATEMENT

C.G.A. is a co-founder of a startup company, Ionomer Solutions LLC, that is in the process of licensing HT-PEM materials (US Patent Application No. 62/656,538) and HT-PEM EHP technology (US Patent Application No. 63/192,607) developed at Louisiana State University with plans for commercialization

## REFERENCES

1. Cullen, D. A.; Neyerlin, K. C.; Ahluwalia, R. K.; Mukundan, R.; More, K. L.; Borup, R. L.; Weber, A. Z.; Myers, D. J.; Kusoglu, A., New roads and challenges for fuel cells in heavy-duty transportation. *Nature Energy* **2021**, *6* (5), 462-474.
2. Lively, R. P., The refinery of today, tomorrow, and the future: A separations perspective. *Aiche Journal* **2021**, *67* (7).
3. Wagner, F. T.; Lakshmanan, B.; Mathias, M. F., Electrochemistry and the Future of the Automobile. *The Journal of Physical Chemistry Letters* **2010**, *1* (14), 2204-2219.
4. Lee, K.-S.; Spendelov, J. S.; Choe, Y.-K.; Fujimoto, C.; Kim, Y. S., An operationally flexible fuel cell based on quaternary ammonium-biphosphate ion pairs. *Nature Energy* **2016**, *1* (9), 16120.
5. Atanasov, V.; Lee, A. S.; Park, E. J.; Maurya, S.; Baca, E. D.; Fujimoto, C.; Hibbs, M.; Matanovic, I.; Kerres, J.; Kim, Y. S., Synergistically integrated phosphonated poly(pentafluorostyrene) for fuel cells. *Nature Materials* **2021**, *20* (3), 370-377.
6. Venugopalan, G.; Chang, K.; Nijoka, J.; Livingston, S.; Geise, G. M.; Arges, C. G., Stable and Highly Conductive Polycation-Polybenzimidazole Membrane Blends for Intermediate Temperature Polymer Electrolyte Membrane Fuel Cells. *Acs Applied Energy Materials* **2020**, *3* (1), 573-585.
7. Venugopalan, G.; Bhattacharya, D.; Kole, S.; Ysidron, C.; Angelopoulou, P. P.; Sakellariou, G.; Arges, C. G., Correlating high temperature thin film ionomer electrode binder properties to hydrogen pump polarization. *Materials Advances* **2021**, *2* (13), 4228-4234.
8. Mack, F.; Aniol, K.; Ellwein, C.; Kerres, J.; Zeis, R., Novel phosphoric acid-doped PBI-blends as membranes for high-temperature PEM fuel cells. *Journal of Materials Chemistry A* **2015**, *3* (20), 10864-10874.
9. Falkenhainer, B.; Forbus, K. D., Compositional modeling: finding the right model for the job. *Artificial Intelligence* **1991**, *51* (1), 95-143.
10. Fisher, O. J.; Watson, N. J.; Escrig, J. E.; Witt, R.; Porcu, L.; Bacon, D.; Rigley, M.; Gomes, R. L., Considerations, challenges and opportunities when developing data-driven models for process manufacturing systems. *Computers & Chemical Engineering* **2020**, *140*, 106881.
11. Wu, Z.; Rincon, D.; Christofides, P. D., Real-Time Adaptive Machine-Learning-Based Predictive Control of Nonlinear Processes. *Industrial & Engineering Chemistry Research* **2020**, *59* (6), 2275-2290.
12. Briceno-Mena, L. A.; Venugopalan, G.; Romagnoli, J. A.; Arges, C. G., Machine learning for guiding high-temperature PEM fuel cells with greater power density. *Patterns* **2021**, 100187.
13. Pan, S. J.; Yang, Q., A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering* **2010**, *22* (10), 1345-1359.
14. Perkins, D. N.; Salomon, G., Transfer of Learning. In *International Encyclopedia of Education*, 2nd ed.; Husen, T.; Postlethwaite, T. N., Eds. Pergamon Press: Oxford, England, 1992.
15. Yang, Q.; Zhang, Y.; Dai, W.; Pan, S. J., *Transfer Learning*. Cambridge University Press: Cambridge, 2020.

16. Zhu, W.; Braun, B.; Chiang, L. H.; Romagnoli, J. A., Investigation of transfer learning for image classification and impact on training sample size. *Chemometrics and Intelligent Laboratory Systems* **2021**, *211*, 104269.
17. Zhu, W.; Romagnoli, J. In *General Feature Extraction for Process Data Using Convolutional Neural Network Based Transfer Learning*, 2020 Spring Meeting & 16th Global Congress on Process Safety, AIChE: 2020.
18. Szegedy, C.; Wei, L.; Yangqing, J.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. In *Going deeper with convolutions*, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 7-12 June 2015; 2015; pp 1-9.
19. He, K.; Zhang, X.; Ren, S.; Sun, J. In *Deep Residual Learning for Image Recognition*, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 27-30 June 2016; 2016; pp 770-778.
20. Bishop, C. M., *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag: 2006.
21. Khirirat, S.; Feyzmahdavian, H. R.; Johansson, M. In *Mini-batch gradient descent: Faster convergence under data sparsity*, 2017 IEEE 56th Annual Conference on Decision and Control (CDC), 12-15 Dec. 2017; 2017; pp 2880-2887.
22. Kingma, D. P.; Ba, J., Adam: A Method for Stochastic Optimization. *CoRR* **2015**, *abs/1412.6980*.
23. Vishwakarma, G.; Haghghatlari, M.; Hachmann, J., Towards autonomous machine learning in chemistry via evolutionary algorithms. *ChemRxiv* **2019**, doi: 10.26434/chemrxiv.9782387.v1.
24. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T., A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* **2002**, *6* (2), 182-197.
25. Blank, J.; Deb, K., Pymoo: Multi-Objective Optimization in Python. *IEEE Access* **2020**, *8*, 89497-89509.

# TOC

