Transfer Learning of High-Fidelity Opacity Spectra in Autoencoders and Surrogate Models

Michael D. Vander Wal, Ryan G. McClarren, and Kelli D. Humbird

Abstract—Simulations of high energy density physics are expensive, largely in part for the need to produce nonlocal thermodynamic equilibrium opacities. High-fidelity spectra may reveal new physics in the simulations not seen with low-fidelity spectra, but the cost of these simulations also scales with the level of fidelity of the opacities being used. Neural networks are capable of reproducing these spectra, but neural networks need data to train them, which limits the level of fidelity of the training data. This article demonstrates that it is possible to reproduce high-fidelity spectra with median errors in the realm of 3%–4% using as few as 50 samples of high-fidelity Krypton data by performing transfer learning on a neural network trained on many times more low-fidelity data.

Index Terms—Deep jointly informed neural network (DJINN), multifidelity, nonlocal thermal equilibrium (NLTE), opacities, transfer learning.

I. INTRODUCTION

INERTIAL confinement fusion (ICF) is currently one of the experimental approaches to controlled nuclear fusion. ICF involves the heating and compressing of a deuterium and tritium (DT) fuel target. This is generally accomplished with lasers that are used to rapidly heat the fuel either directly or indirectly such that shock waves are formed in the fuel, which compresses and heats the fuel further. As indicated by the name of the process, lasers directly illuminate the fuel in direct-drive ICF. In the case of indirect-drive ICF, lasers heat the inside of a hohlraum, which subsequently causes the generation of high-intensity X-rays. The X-rays then heat and compress the fuel capsule, which sits inside of the hohlraum [1], [2], [3].

ICF experiments are exquisitely complex to the field and expensive to execute. Each “shot,” as the experiments are often called, can easily cost in the realms of $1 million. This means, of course, that experiments are not performed lightly, and significant effort goes into designing the experiments using computational models.

Computer simulations are comparatively cheap and are, thus, useful tools for scouting out vast design spaces to search for optimal experimental settings. Simulations of ICF implsions are multiphysics simulations, involving hydrodynamics, radiation transport, nuclear fusion burn, equations of state at extreme conditions, and more. Depending on the degree of accuracy needed in each physics subpackage, approximations can be made, or tabled data may be used to reduce the complexity of an integrated simulation. For example, opacities can be tabulated for assumptions of local thermodynamic equilibrium, but there are regimes in ICF experiments that violate these assumptions. In this case, higher fidelity opacity calculations are necessary to capture important physical processes accurately [4], [5]. In this work, we focus on improving the fidelity of opacity calculations without significantly increasing the computational cost of the overall ICF simulation using machine learning.

The two primary components of opacity are absorptivity and emissivity. At conditions that one might generally encounter throughout everyday life, these properties are considered to be equal and only dependent on temperature and density as prescribed by the blackbody assumption and fall under the realm of local thermal equilibrium (LTE) calculations [3], [4], [5]. ICF conditions, however, demand that non-LTE (NLTE) calculations be made. NLTE calculations are dependent on both the density and temperature of the material in question, as well as the radiation- and electron-energy distributions. Unlike LTE conditions where the relevant radiation interaction parameters can be tabulated and referenced during a calculation, such tables for NLTE cannot be easily constructed, except for in special, limited circumstances [6], [7], or using simplified models [8], [9]. Recent work [10] has developed a parameterization scheme for the electron distributions to simulate NLTE Kα and Kα emission spectroscopy. In addition, two recent papers showed that it is possible to form an economical tabular representation of the NLTE material data based on a linear response model [11] where the table is indexed using a series of radiative fields and their derivatives or using a reduced representation of the radiative field using the bands’ method [12]. The linear response model was also used to create tables used in a hohlraum simulation. These methods are competitors to our approach and are further discussed in the conclusions.
NLTE calculations are significantly more complex than LTE calculations and, thus, have a much higher computational cost. Indeed, this cost is so high that the calculation of the spectral absorptivity and spectral emissivity may constitute as much as 90% of the total computation time of an ICF hohlraum simulation. This cost is dictated by the level of fidelity of the model. The level of fidelity is a function of the fineness of energy-binning and the number and selection of atomic states that are used to compute the spectra. The difference in these levels manifests such that low-fidelity models end up with many spectral lines or spikes grouped together and effectively being smoothed out, as seen in Fig. 1. This is particularly apparent in low-temperature conditions and low-density conditions. The former is effectively the initial condition of the simulation. In some cases, the level of fidelity desired may require a whole day to complete a single calculation [5]. This level of fidelity is not currently used to run full ICF simulations. The fidelity of the models used in the actual ICF simulations is necessarily much lower; however, they are still expensive.

This is where machine learning or more specifically neural networks become exceptionally useful. Neural networks can be considered universal function approximators [13], [14], [15]. Neural networks, once trained, provide an efficient route to calculate any desired function or set of functions. This is because neural networks can provide an answer without the iterative solving of equations. In addition, where some solvers are not well suited for parallelization, neural networks are embarrassingly parallel [16]. Indeed, it has been demonstrated that a 7× speedup was observed for simulations in Hydra where a set of neural networks replaced Cretin [5]. Hydra is the hydrodynamics multiphysics simulation software most commonly used for ICF simulations at the Lawrence Livermore National Laboratory [17]. Cretin is the standard package used by Hydra to calculate the NLTE opacities of materials [18]. It was also demonstrated that multiple elements could be included in a single model [19]. An important finding from that work is that data need to be properly scaled, so the arithmetic inherent to neural networks does not overwhelm the incredibly small values in some atomic spectra. Scaling with \( \log_{10}(x+1) \) works well for values greater than one, but, for values less than one, the output is even smaller than the input. Both of these examples utilized large datasets based on low-fidelity NLTE calculations and do not address the desire to use the time-cost prohibitive high-fidelity models. If this high-fidelity data could be calculated with a neural network, the 7× speedup could very well become a 75× speedup or higher relative to the direct computation. This expectation considers that the low-fidelity models used in the proof-of-concept take less than 1 s to compute a single set of opacity calculations. The challenge with applying this technique to high-fidelity opacity data is that acquiring the same amount of data required for training the low-fidelity network is itself prohibitively expensive.

Transfer learning is a proposed method to reduce the computational time required to train an accurate high-fidelity neural network. Transfer learning is any of various techniques that take an already trained neural network and reuses all or only parts of the neural network to achieve either similar or potentially very different tasks [20]. Transfer learning is also used to compensate for small datasets [21], [22], [23], [24], [25], [26], [27], [28], [29], [30]. This article explores the use of various transfer learning techniques to achieve the goal of predicting high-fidelity opacities with similar accuracy as a low-fidelity opacity model while using a comparatively small dataset. The methods that are explored include simple retraining on the new data, retraining on the new data with various collections of layers being frozen, and linking a pretrained low-fidelity network to an untrained high-fidelity network [22]. The ultimate purpose of this work is to demonstrate that transfer learning can be used to effectively learn high-fidelity opacity models, as well as give direction and suggestions on how to perform this task.

A. Prior Work and Contributions

Transfer learning with autoencoders has been performed for a wide variety of tasks. These often include the use of an autoencoder to learn existing structures. The autoencoder’s decoder is then replaced with a new network that performs the desired task, such as classification [24], [25], [26], [27], [28], [29]. In a few cases, there has been work that has not been directly attributed as being transfer learning but does closely resemble transfer learning as a preexisting network is used [22], [23]. In a manner similar to the use of transfer learning from autoencoders to produce classification networks, the works in [5] and [19] replace the classification networks with regression networks. Once again, they are not directly attributed to the process of transfer learning but are similar in nature. In general, there appears to be less in the way of transfer learning for the purposes of regression [22], [23], [31], [32], [33], [34]. Indeed, some of those cases were not even directly attributed as transfer learning is among the small amount of work done in this area. The work shown in [23] describes the approach taken in this article as a “brute force” approach.
Utilizing neural networks to predict spectra or material properties is not itself new either. There has been much work on predicting the spectra of chemical or elemental species and the inverse problem: the prediction of which chemical or elemental species are present based on observed spectra. Also, in the field of chemistry, transfer learning has been used to improve the performance of autoencoders in generating new chemicals.

The work in this article is an extension of efforts to reproduce spectra—absorptivity and emissivity—for NLTE conditions. In our initial work, autoencoders are trained to compress and decompress the spectra, and then, deep jointly informed neural network (DJINN) models are trained to predict the latent spaces of those autoencoders. During prediction, the DJINN model takes the inputs to produce the latent space representation that the autoencoder's encoder would have produced. The autoencoder's decoder then deconstructs the predicted latent space into the predicted spectra. The contributions of this article are the demonstration that high-fidelity models can be learned by the process of transfer learning from low-fidelity models that median relative errors on the order of 3%–4% can be achieved, and this can be obtained with as few as 50 high-fidelity samples.

II. METHODS

A. Data

The training data are generated using the code Cretin. For both the low-fidelity and high-fidelity calculations, Cretin solves the equations of the collisional-radiative model in steady state using a screened hydrogenic model. The difference between the calculations is that the low-fidelity calculations use 1849 possible electronic configurations of the krypton atom, whereas the high-fidelity calculations use 25,903 levels following Cretin. The contributions of this article are the demonstration that high-fidelity models can be learned by the process of transfer learning from low-fidelity models that median relative errors on the order of 3%–4% can be achieved, and this can be obtained with as few as 50 high-fidelity samples.

In this expression, \( a \) is the radiative constant of \( 7.5657 \times 10^{-15} \text{erg/cm}^3/\text{K}^4 \), \( b(v, T_e) \) is the reduced Planckian, \( g(v) \) is a Gaussian distribution centered at 3 keV with a width of 0.8 (at half-maximum set to 1 keV), and \( \alpha \) is the M-band ratio that gives the fraction of the radiation field in the M-band between 2- and 4-keV characteristics of gold.

Using these randomly generated inputs, 320,000 low-fidelity krypton spectra training samples are generated each for absorptivity and emissivity. For training the neural networks, each separate model has its own 0.8/0.2 train/test split of the 320,000 samples. The 20% used for testing will never be used to train the model that is associated with that specific split.

The high-fidelity training spectra are produced with a more detailed atomic model in Cretin, with sample sizes of 25, 50, 100, 250, 500, 1000, and 2000. The high-fidelity data are sampled using Latin hypercube sampling (LHS) with the same parameter boundaries defined above. LHS is used to sample the parameter space to ensure that the parameter space range is well-sampled when the number of samples is small. To give perspective on the difference in computation cost between the low- and high-fidelity models, the low-fidelity model generates approximately 33 spectra per minute, and the high-fidelity model generates approximately three spectra per minute. An additional set of 60,000 test spectra is generated using the high-fidelity model with random sampling.

B. Basic Retraining

Our work is based on deep neural networks designed to represent complex atomic physics calculations. Neural networks are comprised of a set of intermediate calculations, usually referred to as neurons, which take input data and perform a calculation to produce an output. The connections between the neurons (and how the neurons combine data) are determined through a set of free parameters, the so-called weights and biases of the network, which are determined through a training process where these parameters are set in order to match existing data known as training data. In addition, there are several hyperparameters that the network designer must choose. These include the number and connectivity of the neurons that give the network architecture and activation functions for the neurons. One such architecture that we use heavily is the autoencoder that seeks to find a low-dimensional subspace on which a high-dimensional dataset lives. In addition, the aforementioned DJINN method attempts to automatically select many of the hyperparameters for a given network based on the training data. The methods that we use in this work have been well-described in several recent publications, and the interested reader is encouraged to consult these for additional background.

First, two different autoencoder architectures are trained on low-fidelity data. One architecture is fully connected and is described in Table I, and the other architecture is convolutional, as described in Table II. These autoencoders are trained to compress and decompress the absorptivity and emissivity spectra with separate autoencoders used for each. Separate DJINN models are trained to reproduce the low-fidelity autoencoders’ latent spaces. In total, there are ten complete models for absorptivity and emissivity for each variation of the model used.
The low-fidelity autoencoders are trained on a 0.9/0.1 split during the actual training session. The 10% holdout is used to produce validation cost. This means that each low-fidelity model is trained on 230,400 samples. Prior to training, the spectra are scaled by taking the 18th root of all bins in the spectra. The fully connected autoencoders are trained for 10,000 epochs with a learning rate of 0.001 while using the mean square error (mse) as the loss function. The batch size is 1280 spectra per batch. All of these training details are the same for the convolutional autoencoders with the exception of the learning rate, which is 0.0001.

The DJINN models are then trained to reproduce the latent space of the autoencoders given the inputs associated with the corresponding spectra. DJINN automatically constructs a model using a few hyperparameters. Those parameters are the dropout rate, the maximum number of layers, and the number of models to use in an ensemble. The dropout rate was set to 0.0, the maximum number of layers to 11, and the number of models to 1. By the very nature of DJINN, the actual number of layers is not guaranteed to be equal to the maximum; however, in this case, all models reach the maximum number of layers. DJINN employs MinMax scaling for its inputs and outputs, and it produces models that have layers that all use ReLU activation functions with the exception of the last layer, which uses a linear activation function. DJINN also uses mse as the loss function. The DJINN models are trained for 2000 epochs with the learning rate set to 0.0001 and the batch size of 1280 spectra per batch.

The autoencoders are trained on high-fidelity spectra with various alterations made to the autoencoders. These alterations consisted of freezing different layers of the autoencoders such that they could not be retrained. For clarity, we define each model by the layers that are trainable during the transfer learning step, as it is more compact to state those than the layers that are frozen. Table III summarizes the trainable layer combinations and the labels attributed to each of them. There is also an instance of no layers being frozen and a brand-new (randomly initialized) model produced on just the high-fidelity spectra.

The hyperparameters used for transfer learning differ from those used to train low-fidelity models. The 25, 50, and 100 spectra datasets use batch sizes of 1. The 250, 500, 1000, and 2000 spectra datasets use two-, three-, five-, and ten-spectra batch sizes. There is also no 0.9/0.1 validation split.

The models are trained for 2000 epochs on high-fidelity data using mse as the loss function. The learning rates of both the fully connected and convolutional models are reduced by a factor of ten, making them 0.0001 and 0.00001, respectively.

Each of the autoencoders produced via the different combinations of frozen–unfrozen layers had DJINN models to match that are either new, continued training of the unaltered model, had the last two layers trainable, or had just the last layer trainable. Ten models are produced for each combination of the autoencoders and DJINN models. The DJINN models produced here are trained with the same batch sizes as their associated autoencoders. The models are trained for 1000 epochs on the high-fidelity latent spaces using a learning rate that is similarly a factor of ten smaller at 0.00001. It should be noted that the new DJINN models during this step are also built with a maximum number of layers set to 11, but, because the number of layers actually produced is related to the number of training samples available at construction time, the actual number of layers tends to be much less than 11.

A very important note for the success in the implementation of neural network models is the scaling of the input and output data of the neural network. The data cover a very large range of values, especially the emissivity spectra, in which values drop lower than $10^{-40}$, and the neural networks have an apparent floor around $10^{-4}$. As discussed in detail in our previous work, this large range requires compression of the order of magnitude of the data [19]. The data here are compressed by taking the 18th root of the data. Compression of the range is necessary because, in our neural networks, the large range of data in atomic spectra leads to extremely small values being computed with relatively large values. Our previous work showed that neural networks could not reproduce results that were more than 7 orders of magnitude smaller than the mean of the data when the original $\log_{10}(x+1)$ scaling was used [5, 19]. As previously mentioned, the $\log_{10}x + 1$ scaling leads to small inputs being even smaller leading to them being quickly overwhelmed by larger values.

### III. Results

Before the detailed explanation of the results, a baseline must be set. The baseline for comparison is the low-fidelity model’s ability to reproduce the high-fidelity results. If the transfer-learned models cannot reproduce the high-fidelity results better than the low fidelity, then there is no point...
in trying. This is not an unreasonable question. The low-fidelity atomic models were created to minimize the error in the results when going from a large number of levels in the high-fidelity model to a smaller model. Therefore, in many regions of parameter space, the low- and high-fidelity results may be very similar. It is important to remember, however, that even the low-fidelity atomic physics calculations are much slower than evaluating a trained neural network [5].

The results from comparing transfer learning to the low-fidelity model can be seen in Table IV. The errors reported throughout this article are all percent relative errors and the $R^2$ values. The errors are presented as a spectral error, meaning that the value represents one specific energy bin in specific spectra. This is different from how the error was reported in [5] and [19] where it was reported on a gray approximation of the spectra. The $R^2$ value is used as a measure to indicate how well the shapes of the spectra are reproduced. Unless otherwise specified, the metrics reported, median, 90th percentile, and maximum, are all the median values of those metrics across the ten models produced for a given combination of the autoencoder and the DJINN model. This is the same for the $R^2$ value with the exception that the 90th percentile will be considered from an inverted ordering (i.e., 90% have a greater $R^2$ value as opposed to less value).

Accompanying the low-fidelity results in Table IV are the results of a selected fully connected model combination and a convolutional model combination. They are also accompanied by the actual low-fidelity data associated with the same inputs as the high-fidelity data. The fully connected model is the model made with a “NO FREEZE” fully connected autoencoder trained on 50 samples, and the convolutional model is the model made with a “NO FREEZE” convolutional autoencoder trained on 50 samples as well. Both of the models utilized a “NO FREEZE” DJINN model. The convolutional model performs better than the fully connected, absorptivity models, but, if the median and 90th-percentile errors are ignored, the convolutional models can be realized to perform worse for emissivity. This is especially apparent if the maximum relative error is of particular importance.

There is the unfortunate reality that the minimum $R^2$ value for the absorptivity models is negative, but the one calculated is still a better value than that of the low-fidelity data. The $R^2$ value for the emissivity models may give rise to some concern; however, the primary reason for this is because the values for the energy bins greater than roughly 8 keV have an even more rapid drop off in value for the high-fidelity data than the low-fidelity data. This is just above the range of energy values of primary importance. The emissivity models, otherwise, have very good shape-matching capacity.

Ultimately, these results also show that both the fully connected models and convolutional models outperform their low-fidelity model counterparts in reproducing high-fidelity data. The high-fidelity models roughly halve both the median and the 90th-percentile relative errors. The low-fidelity data already have a relatively good shape match and, in some cases, better errors, but it tends to be at the extrema of spectra where the bulk majority of this increased error appears. In addition, using more samples does not appear to improve errors or shape match much. Using 2000 samples, the 90th-percentile $R^2$ value was 0.9954 for the fully connected emissivity model, and the 90th-percentile $R^2$ of the convolutional absorptivity model was 0.9954. Neither of these values suggests that using more samples will provide meaningful benefits.

The bottommost lines in the plots in Fig. 2 further show how the convolutional models produce the lowest median errors, and if provided with increasing amounts of data, the median error continues to drop. This is in contrast to the fully connected models, represented by the tight clustering of lines in the plot, which increases in median error when more than 500 samples are provided. Ultimately, these lines show that the choice of trainable layers does not particularly matter for a particular class of autoencoder, fully connected versus convolutional. The same goes for emissivity provided that

| Absorptivity | LF-DATA | LF-FC | LF-CONV | TL-FC | TL-CONV |
|--------------|---------|-------|---------|-------|---------|
| Median       | 2.46    | 7.85  | 76.84   | 3.20  | 2.55    |
| 90th Percentile | 22.3    | 41.9  | 40.0    | 20.3  | 19.9    |
| Maximum      | 1.80 x 10^4 | 2.10 x 10^4 | 1.84 x 10^4 | 1.94 x 10^4 | 1.83 x 10^4 |
| $R^2$ 90th Percentile | 0.9973  | 0.8926 | 0.9926  | 0.9988 | 0.9990  |
| $R^2$ Minimum | -0.3733 | -0.2678 | -0.2100 | -0.2547 | -0.1784 |

| Emissivity | LF-DATA | LF-FC | LF-CONV | TL-FC | TL-CONV |
|------------|---------|-------|---------|-------|---------|
| Median      | 3.19    | 8.04  | 7.87    | 4.32  | 3.81    |
| 90th Percentile | 32.8    | 70.9  | 72.7    | 37.0  | 36.9    |
| Maximum     | 5.04 x 10^5 | 4.44 x 10^5 | 1.95 x 10^16 | 2.26 x 10^11 | 7.82 x 10^20 |
| $R^2$ Median | 0.9999  | 0.9979 | 0.9978  | 0.9998 | 0.9998  |
| $R^2$ 90th Percentile | 0.9961  | 0.9909 | 0.9905  | 0.9951 | 0.9944  |
| $R^2$ Minimum | 0.8921  | 0.8778 | 0.8034  | 0.8822 | 0.7619  |
Fig. 2. Results of models made a “NO FREEZE” DJINN model and the various different autoencoder options. The tight clustering of the lines in these plots makes it clear that the freezing of autoencoder layers does not play a clear or major role. However, the bottom-most lines show that convolutional autoencoders perform better. The vertical lines are the 90th-percentile error. The error values reported are the median values of each reported metric (median or 90th percentile) out of ten separate models.

there are enough samples. The top two lines of the plots in 2 are from the models with a transfer learned DJINN model but a brand-new autoencoder. The significantly worse performance for these two models can be explained by the fact that the DJINN model was originally trained on a different latent space because the autoencoder is different.

One last interesting thing to note is that, for absorptivity and, to a more limited extent, emissivity, using more samples does not guarantee better median errors or even maximum errors.

The entirely brand-new models performed similar to the models that had a brand-new DJINN model made with a transfer-learned autoencoder. The results of a brand-new DJINN model built from the various autoencoders can be seen in Fig. 3. Building the wholly brand-new models, i.e., the “NO TRANSFER,” model. This result provides a demonstration that it is the DJINN model that is the primary contributing factor to the error.

Fig. 3. These plots are the results of the models that utilized a brand-new, i.e., the “NO TRANSFER,” model. This result provides a demonstration that it is the DJINN model that is the primary contributing factor to the error.

learned autoencoders; Fig. 3 shows that all of the models follow the same basic curve. That being said, maximum error performance does generally get better with increased available data, and the brand-new models do not perform nearly as well in maximum error than the transfer-learned models. The convolutional models still tend to perform better than fully connected networks for absorptivity, but this is not always true for emissivity.

The results of a brand-new DJINN model built from a “NO FREEZE” autoencoder can be seen in Fig. 4, which compares it against the DJINN models built from “NO FREEZE” autoencoders. It is clear that, for both absorptivity and emissivity, the brand-new DJINN models do not begin to also perform the fully transfer-learned models until about 1000 samples are used. A note to make about the brand-new models is that the models made with <1000 samples produce predictions that are the correct shape, but those predictions have a large offset from the target data.

These results justify that, based purely on error, transfer learning will be needed if only small amounts of data are
available. On a positive note, it does show that only about 1000 samples are necessary to build an acceptably accurate model.

This is more apparent in the absorptivity spectra than the emissivity spectra despite absorptivity having the least amount of error. Most of higher density (> 0.01 g/cm³) inputs produce spectra that are only a few percent difference between the low-fidelity and high-fidelity data, and the shape is generally very similar for emissivity. The low-density inputs produce high-fidelity spectra that are significantly different from the low-fidelity data, as seen in Fig. 1.

The error associated with shape is difficult to quantify, but it is inherently embedded in the general numerical error. Thus, a representation of the distribution of errors can be seen in Fig. 5. The heat maps show that both absorptivity and emissivity are primarily dependent on the density with the emissivity appearing to be more so than the absorptivity. This conclusion is based on the less smooth gradient in the heat map for absorptivity. The red dots in the heat maps show that the worst spectral reproductions are concentrated in the lowest temperature (<500 eV) and highest temperature (>2000 eV) combined with the lowest density (<.005g/cc) regions where the density appears to be the most contributing factor.

In Fig. 5, we notice that the largest errors occur toward the edge of the input ranges for density and temperature. Not depicted in the figure is the other input to the NLTE calculation and the radiative field. The primary reason for the largest errors being at the edges of the training data is a mathematical one: data-driven models (such as NNs) typically perform worst near the edges of the training data. There is also a physics reason for the large error samples to exist at the low-density portion of the figure. At lower densities, fewer collisions between atoms cause the atomic to have more detailed absorptivity and emissivity curves. These types of spectra will naturally be harder to reproduce. It is also worth noting that our models give the best results at relatively low density (below 0.1 g/cm³) and at higher temperatures (>1 keV). These conditions should require NLTE physics to describe the radiation interactions. In a radiation hydrodynamics calculation, it may be beneficial to switch to an LTE model when conditions justify that approximation; this would also require considering the local radiative field conditions. Our results do not indicate that this is necessary to do as the ML models have a behavior that is more
stable as the density is increased and can handle variations in the radiative field.

The smooth line of the low-fidelity data is effectively a line of best fit between all of the peaks and troughs of the high-fidelity data. This probably produces a “deep” local minimum in the model that training cannot be escaped under the same conditions that place the model in that minimum.

As with most machine learning tasks, this approach works well, but it is not perfect. The spectra that demonstrate the greatest difference between low fidelity and high fidelity prove to be the more difficult spectra to reproduce. However, the bulk majority of spectra are reproduced well. Fig. 6 shows the outcome of the 90th percentile of the \((1 - R^2)\) value. These plots also demonstrate the true effectiveness of transfer learning in that the “Brute NT” (brute force “NO TRANSFER”) model has a relatively constant offset for absorptivity. The “Brute NF” (“NO FREEZE”) model does not possess this offset. The “Brute NT” model performs better for emissivity, but it is still outdone by the transfer learned model.

The errors associated with energy bins where the largest differences in features can also be seen in the overall binwise errors. Fig. 7 provides a visualization of where the errors are concentrated among the energy bins. By comparing the plots in Fig. 6 with the plots in Fig. 7, one can see that the regions of the poor shape match appear to be associated with jumps in error, which also happens to be where either the spectral peaks or spectral troughs are located. For many of the spectra, the biggest difference between low fidelity and high fidelity is the height of the peaks, which is not much, but there are certain situations where the peak actually shifts. The peak in error located around 8–9 keV is the result of one of those shifts. This particular case was that the spectral peak needs to shift to a lower energy bin.

During the exploration of various methods not mentioned, other information was gleaned during additional attempts to add an intermediate step between the low- and high-fidelity data; it was found that the intermediate fidelity data, which normally takes about a quarter of the time to compute, are almost identical to the high-fidelity data. They are so similar that the lines can hardly be distinguished from each other.
to the R2 value, may correspond to spectra with a shape based on the median relative error of the spectra as opposed to the actual spectra of the worst prediction, set that is closest in the Euclidean distance to the inputs on a plot. Furthermore, the set of inputs in the training that is significantly different from the desired shape. Such an example of this can be seen in Fig. 8 where the closest intermediate-fidelity absorptivity spectra match the shape of the expected absorptivity spectra closely, but the closest high-fidelity spectra are significantly different from the expected spectra. In the exact opposite manner, the closest intermediate-fidelity emissivity is significantly different from the expected emissivity spectra, but the closest high-fidelity spectra do look similar. The difference in the case of the emissivity model here is that the model actually did a comparatively decent job at matching the spectra’s shape. This leads the authors to conclude that there are not enough data points or that random sampling with so few samples is not effective.

### IV. Conclusion

In this article, it has been demonstrated that transfer learning high-fidelity data from a low-fidelity model using only 50 high-fidelity samples can achieve median relative errors in the realm of 3%–4%. The improvement over using a model trained on only low-fidelity data is about 2×. Also, simply having more samples does not guarantee significantly better error values. It also happens that the barrier to achieving even better error values may very well be the difficulty in matching the shape of the spectra. The following suggestions are based on the authors’ experience and the results reported in this article.

The authors’ suggest that, as should be expected, the highest level of fidelity that can be reasonably produced in quantities exceeding 10000 samples should be used as the initial low-fidelity data. The reason for this is that the low-fidelity data here were so smooth that it was most likely the biggest obstacle to achieving better spectral shape reproduction than was seen. Next, if the numerical approximation is all that is desired, 50 samples of high-fidelity data appear to be all that is necessary to transfer learn the low-fidelity models to reproduce high-fidelity data.

In terms of the method to be used, a low-fidelity model should first be produced from 10000 or more samples with the temperature being sampled from a uniform distribution and the density being sampled from the uniform distribution on the log10 transform of the density. A fully connected network should be used for emissivity, and a convolutional model should be used for absorptivity. If using it as a convolutional model for emissivity, it should be noted that it will almost certainly produce maximum errors significantly higher than a fully connected network. However, this error will be located in the high-energy region of the spectra where values are extremely low and in a region that may not be of particular concern. In either case, the selection of which layers are frozen does not appear to matter that much; thus, for the sake of simplifying implementation, it is suggested that all layers be kept trainable.

Perhaps, just as important as the suggestions provided above is an intimate knowledge of the direct computation models. As discovered with the intermediate-fidelity data briefly mentioned in regards to 8, there is almost no difference between it and the high-fidelity data, yet the high-fidelity data require 4× the amount of time to compute. The difference in the calculated values is well within the expected range of error that might be seen in the approximation by neural network models; however, this may not hold for all binning structures. It is suggested that, in regions of the input space where there is very little difference in the spectra, lower fidelity data are used to directly augment the training dataset effectively making the training data much cheaper to produce. The point at which this should be avoided can only be determined by the user’s desired level of error.

Finally, to improve upon this work, an exploration of whether the premise of only needing 50 samples will hold for different binning structures and resolutions. Furthermore, because this exploration of transfer learning was only done for a Cretin surrogate model, this work will have its implementation in actual Hydra simulations tested. Hydra models use actual radiative fields from simulations that need to be compressed with an autoencoder. Finally, it must be determined whether there is any discernible difference in the outcomes of the simulations.

The tabular NLTE approach based on the linear response method [11] and the bands’ method [12] should be compared with our approach in future work. Whereas the radiative fields used in the tables are, necessarily, based on simplified representations, the machine learning approach can be trained with a more complicated representation of the radiative field. It is likely that the tabular approach is faster with the ML method having higher accuracy because it can be trained on a wider variety of radiative fields. Furthermore, the physics insight represented in those two methods could also be used to refine future machine learning models.
ACKNOWLEDGMENT

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

REFERENCES

[1] J. Kline and et al., “Progress of indirect drive inertial confinement fusion in the United States,” Nucl. Fusion, vol. 59, Jul. 2019, Art. no. 112018.
[2] Y. Li, W. Huo, and K. Lan, “A novel method determining the M-band fraction in laser-driven gold hohlraums,” Phys. Plasmas, vol. 18, Feb. 2011, Art. no. 022701.
[3] E. S. Dodd and et al., “Hohlraum modeling for opacity experiments on the national ignition facility,” Phys. Plasmas, vol. 25, Jun. 2018, Art. no. 063301.
[4] D. A. Holladay, C. J. Fontes, W. P. Even, and R. G. McClaren, “An accelerated approach to inline non-LTE modeling,” High Energy Density Phys., vol. 34, Mar. 2020, Art. no. 100746.
[5] G. Khut et al., “Deep learning for NLTE spectral opacities,” Phys. Plasmas, vol. 10, May 2020, Art. no. 052707.
[6] H. A. Scott, “Non-LTE radiation transport in high radiation plasmas,” High Energy Density Phys., vol. 1, no. 1, pp. 31–41, Dec. 2005.
[7] C. J. Fontes, J. Abdallah, R. E. H. Clark, and D. P. Kilcrease, “Non-LTE modeling of gold plasmas,” J. Quant. Spectrosc. Radiat. Transf., vol. 65, nos. 1–3, pp. 223–230, Apr. 2000.
[8] H. Takabe and T. Nishikawa, “Computational model for non-LTE atomic process in laser produced plasmas,” J. Quant. Spectrosc. Radiat. Transf., vol. 51, no. 1–2, pp. 379–395, Jan. 1994.
[9] S. X. Hu, G. Fiksel, V. N. Goncharov, S. Skupsy, D. D. Meyerhofer, and V. A. Smalyuk, “Mitigating laser imprint in direct-drive inertial confinement fusion implosions with high-Z dopants,” Phys. Rev. Lett., vol. 108, no. 19, May 2012, Art. no. 195003.
[10] T. Walton et al., “Parameterizing hot electron energy distributions for tabular emissivities and opacities,” High Energy Density Phys., vol. 35, Jun. 2020, Art. no. 100730.
[11] H. A. Scott, J. A. Harte, M. E. Foord, and D. T. Woods, “Using tabulated NLTE data for hohlraum simulations,” Phys. Plasmas, vol. 29, no. 8, Aug. 2022, Art. no. 082705.
[12] Y. Frank and H. A. Scott, “The bands method for tabulating NLTE material properties,” High Energy Density Phys., vol. 43, Jun. 2022, Art. no. 100998.
[13] F. Scarselli and A. C. Tsoi, “Universal approximation using feedforward neural networks: A survey of some existing methods, and some new ideas,” Neural Netw., vol. 11, no. 1, pp. 15–37, Jan. 1998.
[14] G. Cybenko, “Approximation by superpositions of a sigmoidal function,” Mathematika Control, Signals, Syst., vol. 2, pp. 303–314, Dec. 1989.
[15] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” Neural Netw., vol. 2, no. 5, pp. 359–366, Jan. 1989.
[16] R. D. Prabhu, “Gneuron: Parallel neural networks with GPU,” in Proc. Int. Conf. High Perform. Comput., 2009.
[17] M. M. Marinak et al., “Three-dimensional hydra simulations of national ignition facility targets,” Phys. Plasmas, vol. 8, no. 5, pp. 2275–2280, 2001.
[18] H. A. Scott, “Cretin—A radiative transfer capability for laboratory plasmas,” J. Quant. Spectrosc. Radiat. Transf., vol. 71, nos. 2–6, pp. 689–701, Oct. 2001.
[19] M. D. Vander Wal, R. G. McClaren, and K. D. Humphid, “Neural network surrogate models for absorptivity and emissivity spectra of multiple elements,” Mach. Learn. with Appl., vol. 8, Jun. 2022, Art. no. 100308.
[20] R. G. McClaren, Machine Learning for Engineers: Using Data to Solve Problems for Physical Systems. Springer, 2021. [Online]. Available: https://books.google.com/books?id=RLhyszgEACAAJ
[21] F. Zhuang et al., “A comprehensive survey on transfer learning,” Proc. IEEE, vol. 109, no. 1, pp. 43–76, Jul. 2021.
[22] X. Meng and G. E. Karniadakis, “A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse PDE problems,” J. Comput. Phys., vol. 401, Jan. 2020, Art. no. 109020.
[23] R. C. Aydin, F. A. Braeu, and C. J. Cyron, “General multi-fidelity framework for training artificial neural networks with computational models,” Frontiers Mater., vol. 6, p. 61, Apr. 2019.
[24] W. Luo, J. Li, J. Yang, W. Xu, and J. Zhang, “Convolutional sparse autoencoders for image classification,” IEEE Trans. Neural Netw. Learn. Syst., vol. 29, pp. 1–6, 2016.
[25] Y. Zhu, X. Wu, P. Li, Y. Zhang, and X. Hu, “Transfer learning with deep manifold regularized auto-encoders,” Neurocomputing, vol. 369, pp. 145–154, Dec. 2019.
[26] F. Zhuang, X. Cheng, P. Luo, S. J. Pan, and Q. He, “Supervised representation learning: Transfer learning with deep autoencoders,” in Proc. 24th Int. Joint Conf. Artif. Intell. (IJCAI), 2015, pp. 1–7.
[27] J. Deng, Z. Zhang, E. Marchi, and B. Schuller, “Sparse autoencoder-based feature transfer learning for speech emotion recognition,” in Proc. Humaine Conf. Affect. Comput. Intell. Interact., 2013, pp. 1–6.
[28] J. Deng, R. Xia, Z. Zhang, Y. Liu, and B. Schuller, “Introducing shared-hidden-layer autoencoders for transfer learning and their application in acoustic emotion recognition,” in Proc. IEEE Int. Conf. Acoustic, Speech Signal Process. (ICASSP), May 2014, pp. 4818–4822.
[29] J. M. Eusebio, “Learning transferable data representations using deep generative models,” M.S. thesis, Arizona State Univ., 2018.
[30] T. Imai, S. Watanabe, and H. Nobuhara, “Stepwise PathNet: A layer-by-layer knowledge-selection-based transfer learning algorithm,” Sci. Rep., vol. 10, no. 1, pp. 1–14, May 2020.
[31] B. L. Kustowski, J. A. Gaffney, B. K. Spears, G. J. Anderson, J. J. Thiagarajan, and R. Anirudh, “Transfer learning as a tool for reducing simulation bias: Application to inertial confinement fusion,” in Proc. 9th Annu. Meet. APS Division Plasma Phys., 2019.
[32] S. Goswami, C. Antunescu, S. Chakraborty, and T. Rabiczuk, “Transfer learning enhanced physics informed neural network for phase-field modeling of fracture,” Theor. Appl. Fract. Mech., vol. 106, Apr. 2020, Art. no. 102447.
[33] K. D. Humbird, J. L. Peterson, J. Salmonson, and B. K. Spears, “Cognitive simulation models for inertial confinement fusion: Combining simulation and experimental data,” 2021, arXiv:2105.0099.
[34] K. D. Humbird, J. L. Peterson, B. K. Spears, and R. G. McClaren, “Transfer learning to model inertial confinement fusion experiments,” IEEE Trans. Plasma Sci., vol. 48, no. 1, pp. 61–70, Jan. 2020.
[35] K. Ghosh et al., “Deep learning spectroscopy: Neural networks for molecular excitation spectra,” Adv. Sci., Mach. Learn., vol. 6, May 2019, Art. no. 1801367.
[36] M. Chatzidakis and G. A. Botton, “Towards calibration-invariant spectroscopy using deep learning,” ISPRS Int. J. Geo-Inf., vol. 8, no. 4, p. 160, Mar. 2019.
[37] M. Chatzidakis and G. A. Botton, “Towards calibration-invariant spectroscopy using deep learning,” ISPRS Int. J. Geo-Inf., vol. 8, no. 4, p. 160, Mar. 2019.
Michael D. Vander Wal received the B.S. degree in mechanical engineering from the South Dakota School of Mines and Technology, Rapid City, SD, USA, in 2018, and the M.S. degree in mechanical engineering and the Ph.D. degree in aerospace and mechanical engineering from the University of Notre Dame, Notre Dame, IN, USA, in 2020 and 2022, respectively.

Kelli D. Humbird received the Ph.D. degree in nuclear engineering from Texas A&M University, College Station, TX, USA, in 2019. She is currently a Physicist with the Lawrence Livermore National Laboratory, Livermore, CA, USA. She is a Researcher within the Inertial Confinement Fusion Program, where her scientific contributions include predictive modeling with machine learning and statistical methods, uncertainty quantification, hohlraum design, data analytics, and high-frequency computing.

Ryan G. McClarren received the Ph.D. degree from the University of Michigan at Ann Arbor, Ann Arbor, MI, USA, in 2007. He is currently an Associate Professor of aerospace and mechanical engineering with the University of Notre Dame, Notre Dame, IN, USA. Prior to joining the University of Notre Dame in 2017, he was an Assistant Professor of nuclear engineering with Texas A&M University, College Station, TX, USA, and a Research Scientist with the Computational Physics and Methods Group, Los Alamos National Laboratory, Los Alamos, NM, USA. He has authored numerous publications in refereed journals on machine learning, uncertainty quantification, and numerical methods, and three scientific texts: Machine Learning for Engineers, Uncertainty Quantification and Predictive Computational Science: A Foundation for Physical Scientists and Engineers, and Computational Nuclear Engineering and Radiological Science Using Python. His work applies simulation to understand, analyze, and optimize engineering systems.

Dr. McClarren is a fellow of the American Nuclear Society. He was recently named the Editor-in-Chief of the Journal of Computational and Theoretical Transport.