Neural network representability of fully ionized plasma fluid model closures

Romit Maulik,1 Nathan A. Garland,2 Xian-Zhu Tang,2 and Prasanna Balaprakash3,1

1) Argonne Leadership Computing Facility, Argonne National Laboratory, Lemont, IL 60439, USA
2) Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87545, USA
3) Mathematics and Computer Science Division, Argonne National Laboratory, Lemont, IL 60439, USA

(Dated: February 20, 2021)

The closure problem in fluid modeling is a well-known challenge to modelers aiming to accurately describe their system of interest. Over many years, analytic formulations in a wide range of regimes have been presented but a practical, generalized fluid closure for magnetized plasmas remains an elusive goal. In this study, as a first step towards constructing a novel data based approach to this problem, we apply ever-maturing machine learning methods to assess the capability of neural network architectures to reproduce crucial physics inherent in popular magnetized plasma closures. We find encouraging results, indicating the applicability of neural networks to closure physics but also arrive at recommendations on how one should choose appropriate network architectures for given locality properties dictated by underlying physics of the plasma.
I. INTRODUCTION

In modeling and simulation of plasma physics dynamics many practitioners employ a fluid, or moment, model\textsuperscript{1–3}. The attraction of a fluid model lies in the generally simpler model description, and far reduced computational cost, when compared to high fidelity kinetic or particle based methods that describe a charged particle distribution function $f(r, v, t)$. To determine which of these two approaches to take, one must assess the trade-off between an averaged macroscopic description of a plasma that is computationally tractable, versus a fine kinetic microscopic description that may carry extreme computational cost. In the event of employing a fluid model, one must sufficiently truncate or close the infinite hierarchy of equations in the moments of $f(r, v, t)$.

While truncation at higher order may be done, it is more common to try to enforce some underlying physical properties of the problem to derive an approximate closure for higher order variables, such as heat flux, as a function of the lower order variables found in the model’s conservation equations, such as density, velocity or energy. The physical properties used to derive closures can range from collisional to collisionless regimes, and can enforce local or non-local space-time relationships within the plasma\textsuperscript{4}. This being said, there are some scenarios where complex physics prevents a simple closure being assumed, and the question as to what closure to employ has a non-trivial answer. In this situation, some research communities, particularly fluid mechanics in recent years\textsuperscript{5–7}, have turned to machine learning to try to construct surrogate closure models that map the known macroscopic variables in a fluid model to the higher order moments that must be closed.

Machine and deep learning methods have long been used in fusion and plasma physics applications\textsuperscript{8–12}, and in this study we seek to apply the methodology, inspired by the fluid mechanics community, of using machine learning to formulate and study surrogate models for the aforementioned magnetized plasma closure models that are crucial to fluid modeling. We note recent work of Ma et al.\textsuperscript{13} on the analysis of closure surrogates via neural networks for the Hammett-Perkins closure\textsuperscript{14}. In this work we will also touch on the Hammett-Perkins non-local closure but also expand to include additional closures in different regimes relevant to fusion applications\textsuperscript{2,15}.

In seeking to learn surrogate closure models for use in a fluid model, we aim to provide some physical intuition of the problem \textit{a priori} before setting an arbitrary complex neural network on the problem. This style of approach has gained recent focus as physics-informed machine learning\textsuperscript{16,17}, and we believe this problem can benefit greatly from this philosophy. With this in
mind, the primary goal of this work was to verify the representability, and architecture style, of neural networks to sufficiently reproduce the often non-trivial physics that underpin the formulation of various analytic closure expressions.

We believe that benchmarking learned surrogates against well known analytic closures in important physical limits, and understanding the structure and limitations of the neural networks that formulate closure surrogates, is an instructive exercise. Particularly, it is useful to understand these limits before undertaking large-scale kinetic or particle simulations for data generation, assimilation, and subsequent closure learning in a haphazard manner. Further, from this study we can observe that, in the important physical limits grounding the closures studied, there is a benefit to tailoring a specific network architecture informed by the physics of the plasma regime each closure is designed for, rather than carelessly applying an unnecessarily complex general network architecture.

Our methodology and results are presented as follows. In Section II we briefly review fluid models for magnetized plasmas and some of the closure methods used. Section III outlines the learning method employed to formulate the closure surrogate neural networks, with Section IV detailing the network architectures employed for each surrogate. The results obtained from training the appropriate networks on each benchmark closure are discussed in Section V followed by a summary of the key findings of this work in Section VI.

II. PLASMA FLUID MODELING

The fundamental description of electron dynamics is carried by the electron distribution function, \( f_e(r, v, t) \), in 7D phase space. While many techniques exist to resolve \( f_e(r, v, t) \), the computational load can be enormous. The primary approach is solution of the kinetic equation for \( f_e(r, v, t) \)

\[
\frac{\partial f}{\partial t} + v \cdot \frac{\partial f}{\partial x} + \frac{q}{m_e} (E + v \times B) \cdot \frac{\partial f}{\partial v} = C(f),
\]  

where \( C(f) \) is the collision operator.

As an alternative, a fluid description of the plasma is a popular approach that can simulate plasma properties in a more computationally efficient manner. Macroscopic fluid quantities, such
density $n_e$, mean velocity $u_e$, and pressure $P_e$ are obtained via velocity moments of $f_e(r,v,t)$

$$n_e(r,t) = \int f_e(r,v,t) \, d^3v,$$  \hspace{1cm} (2)

$$u_e(r,t) = \frac{1}{n_e(r,t)} \int v f_e(r,v,t) \, d^3v,$$ \hspace{1cm} (3)

$$P_e(r,t) = m_e \int w u f_e(r,v,t) \, d^3v,$$ \hspace{1cm} (4)

where $w = v - u_e$ is the fluctuation velocity, $m_e$ is the electron mass, and the scalar pressure, $p_e$ often formed as a variable in continuity equations, can be extracted from the pressure tensor via $P_e = p_e I + \Pi_e$ where $I$ is the identity tensor and $\Pi_e$ is the viscosity tensor.

While fluid modeling is an attractive option to simulate plasmas, there are inherent challenges and disadvantages that this technique brings with it. Fundamentally, no exact knowledge of the velocity space distribution of electrons leads to an inherent inaccuracy in modeling discharges far from equilibrium, where a Maxwellian distribution function is often assumed. An extension of not knowing the velocity space distribution is the closure problem, where higher order moments, such as the viscosity tensor, $\Pi_e$, or heat flux, $q_e$, that appear in the derived continuity equations must be approximated in some way to close, or truncate, the system of equations to enable computation.

A. Collisional limit: Braginskii closure

In his seminal review paper\(^2\) Braginskii tackled the problem of showing convincingly that hydrodynamics reigns in a plasma made up of electrons and a single ion species moving through a strong magnetic field, where collisions occur at a rate that is comparable to the gyrofrequency. The well-known Braginskii\(^3\) equations for electron transport, dropping the subscript for $n$, $u$, and $p$, can be written

$$\frac{dn}{dt} + n \nabla \cdot u = 0,$$ \hspace{1cm} (5)

$$m_e n \frac{du}{dt} + \nabla p + \nabla \cdot \Pi + en (E + u \times B) = F,$$ \hspace{1cm} (6)

$$\frac{3}{2} \frac{dp}{dt} + \frac{5}{2} p \nabla \cdot u + \Pi : \nabla u + \nabla \cdot q = W,$$ \hspace{1cm} (7)

where $\frac{d}{dt} = \frac{\partial}{\partial t} + u \cdot \nabla$ is the commonly used convective or material derivative, $\Pi$ is the viscosity tensor, $q$ is the heat flux vector, $F$ is a frictional force due to plasma resistivity and thermal forces, and $W$ is an energy transfer function due to collision with ions and work done by frictional forces.
Braginskii showed, through an asymptotic solution of the kinetic equation underlying equations (5) - (7), one can estimate the heat flux vector $q$, the viscosity tensor $\Pi$, the frictional force vector $F$, and the energy exchange function $W$ in terms of the hydrodynamic observables.

In this study we seek to identify a network architecture that is capable of representing Braginskii’s approximation to the closure given a range of macroscopic input profiles. To that end, a suitable litmus test is to assess the representability of one characteristic piece of the closure functional, namely the so-called gyroviscous stress tensor

$$\Pi_{BG}(n, u, T)(x) = \frac{n(x)T(x)}{4\|B(x)\|} \left( \tilde{\Pi} + \tilde{\Pi}^T \right),$$

where

$$\tilde{\Pi} = \hat{b}(x) \times (\nabla u(x) + \nabla u^T(x)) \cdot (I + 3\hat{b}(x)\hat{b}(x)),$$

where $\hat{b} = B/\|B\|$ is the magnetic field unit vector and $I$ is the $3 \times 3$ identity tensor.

The Braginskii formulation provides one benchmark limit for learned neural network architectures that could provide a surrogate closure model in the collisional magnetized plasma limit, with a local functional dependence. In contrast to this, the following sections now describe non-local closures in collisionless plasma closure limits.

B. Collisionless limit: Hammett-Perkins closure

Linear Landau damping is an inherently kinetic effect in weakly-collisional plasmas wherein fluid moments of the single-particle distribution function decay due to the development of increasingly-fine-scale structure in velocity space. These small-scale features are produced as a result of higher-velocity parts of the distribution function being transported by the streaming effect more rapidly than lower-velocity parts. A similar effect to Landau damping can be observed in sheared flows of conventional fluids\(^\text{18}\).

While the Landau damping effect is kinetic in nature, some aspects of the phenomenon can be mocked-up within a fluid model using an idea originally due to G. W. Hammett and F. W. Perkins.\(^\text{14}\) Their basic idea was to develop a closure for the heat flux in terms of the temperature such that the decay rate predicted by linear Landau damping theory is reproduced exactly by the fluid model when linearized. In a single space dimension the Fourier-space expression for this
closure is given by the simple formula
\[ \hat{q}_k = -n_0 \sqrt{\frac{8}{\pi}} \nu_{th} i \text{sgn}(k) \hat{T}_k. \] (10)

Here \( n_0 \) is the unperturbed plasma density and \( \nu_{th} \) is the nominal plasma thermal speed. In a periodic domain the formula still makes sense provided the wave vector \( k \) is suitably quantized.

Up to an unimportant proportionality factor the formula (10) says that the heat flux is related to the temperature by the famous Hilbert transform from signal analysis and elsewhere in plasma physics\(^\text{[19][20]}\). In the signals context the transform is used, for example, to construct complex representations of signals that are analytic in the upper half-plane, while in the plasma context the transform is required to find the action-angle variables for the continuous spectrum of the linearized Vlasov-Poisson system. If \( H \) denotes the Hilbert transform, and we neglect the unimportant constants, the closure relationship may be written \( q(x) = H[T](x) \). There are a number of well-known formulas for the Hilbert transform including
\[ H[T](x) = \frac{1}{\pi} P \int_{-\infty}^{\infty} \frac{T(s)}{x-s} ds, \] (11)

where \( P \) denotes the principal value, and
\[ H[T](x) = -\frac{1}{\pi} \lim_{\varepsilon \to 0} \int_{\varepsilon}^{\infty} \frac{T(x+s)-T(x-s)}{s} ds, \] (12)

each of which explicitly displays the spatially-global nature of the closure. However where the Fourier space expression (10) needs only minor modification in a periodic domain (a quantization condition on \( k \)) the previous two expressions require more significant changes. For instance the principal value formula (11) on a domain with period \( 2\pi \) becomes
\[ H[T](x) = \frac{1}{2\pi} P \int_{0}^{2\pi} T(s) \cot \left( \frac{x-s}{2} \right) ds. \] (13)

In this study we seek to identify a network architecture that is capable of representing the Hammett-Perkins closure over a domain of temperature profiles sampled randomly from a Gaussian distribution with specified mean \( \bar{T}(x) = \langle T(x) \rangle \) and covariance \( \tilde{T}(x,x') = \langle (T(x) - \bar{T}(x))(T(x') - \bar{T}(x')) \rangle \). That is, we treat the temperature as a Gaussian random field. A similar exercise was also carried out recently using a different training data generation method by Ma et al\(^\text{[13]}\). Like this earlier reference we will focus on learning the closure in configuration space in order to mask its trivial nature in Fourier space. In contrast to the earlier study our analysis will focus on the important issue of extrapolation errors inherent to artificial neural networks. We frame this extrapolation
problem concretely as follows. If a neural network surrogate model for the Hammett-Perkins closure is trained using data sampled with mean $\bar{T}_1(x)$ and covariance $\tilde{T}_1(x,x')$ then how well does the learned closure perform on testing data sampled with mean $\bar{T}_2(x)$ and covariance $\tilde{T}_2(x,x')$? In any future application of neural networks to the problem of closure learning for plasma physics the limitations of the learning process set by extrapolation errors will be crucial to understand and manage.

C. Collisionless limit: Guo-Tang closure

In a collisionless, long $\lambda_{\text{mfp}}$ magnetized plasma, $C(f) = 0$, significant temperature anisotropy can develop. Further, if the plasma under consideration experiences an open magnetic field line, in contrast to the scenario for Hammett and Perkins closure\textsuperscript{[14]}, it can be demonstrated that parallel heat flux can flow from low to high $T_\parallel$ in the vicinity of absorbing boundary walls, contradicting the classical thermal conduction description of heat flux i.e. $q \sim \kappa \nabla T$. In this scenario, a closure derived by Guo and Tang\textsuperscript{[15]} has been demonstrated to sufficiently capture the behavior of the parallel heat flux in the presence of an absorbing boundary with open magnetic field line transport. In the Guo-Tang result closure expressions for both the electron and ion parallel heat flux components are presented, however in this study we solely examine the more physically interesting electron result.

The Guo-Tang closure is applicable to electron transport within an open field line plasma bounded by absorbing walls, where a steady-state is achieved via balance with an upstream electron source plane with a Maxwellian distribution. Throughout the domain electrons experience an open magnetic field, $B$, and ambipolar electric potential, $\phi$. A schematic example of this scenario is shown in Figure 1 indicating source flux, $\Gamma_s$, and temperature $T_s$, wall potential $\phi_w$, as well as magnetic field magnitude at the source, $B_s$, and wall $B_w$.

This closure relates the parallel heat flux, $q_\parallel$, to contributions from parallel, $q_n$, and perpendicular, $q_s$, thermal energies

$$q_\parallel = (q_n + 2q_s) \hat{b}. \quad \text{(14)}$$

The parallel, $q_n$, and perpendicular, $q_s$, contributions to the total heat flux are given by

$$q_n = \left\langle n e v_{ei}^3 \right\rangle - 3n_e u_e T_e \parallel, \quad \text{(15)}$$
$$q_s = \left\langle \mu_e B v_{ei} \right\rangle - n_e u_e T_e \perp, \quad \text{(16)}$$
Figure 1. Geometry of Guo-Tang closure scenario. A symmetric flux expander where the plasma is uniform in the $z$ direction (out of plane) and symmetric about the source plane. Source parameters, $\Gamma_s$ and $T_s$, wall potential, $\phi_w$, and magnetic field magnitudes, $B_s$ and $B_w$, are employed in non-local closure.

where the $n_e u_e T_e$ components represent convective fluxes available from the moment variables used in computation of the fluid model. The crux of the heat flux closure is in the evaluation of both $\langle m_e v_{e\parallel}^3 \rangle$ and $\langle \mu_e B v_{e\parallel} \rangle$, given by

$$
\frac{1}{2} \langle m_e v_{e\parallel}^3 \rangle = \frac{\Gamma_s T_s}{2R^2} \left[ D + R \left( \frac{3}{2} - \frac{1}{R} + \frac{e\phi_{\parallel}}{T_s} \right) \text{erfc} \left( \sqrt{\phi_w} \right) + C \left( R\alpha \frac{e\phi}{T_s} + RG - G - \frac{1}{2} \right) \text{erfi} \left( \sqrt{\alpha\phi_w} \right) \right],
$$

$$
\langle \mu_e B v_{e\parallel} \rangle = \frac{\Gamma_s T_s}{2R^2} \left[ -D + \text{erfc} \left( \sqrt{\phi_w} \right) + C \left( G + \frac{1}{2} \right) \text{erfi} \left( \sqrt{\alpha\phi_w} \right) \right],
$$

where $R = B_s/B$, $e$ is the elementary charge, erfc() and erfi() are the complementary and imaginary error functions, and constants are defined as $\alpha = B_w/(B_s - B_w)$, $\tilde{\phi}_w = e|\phi_w|/T_s$, $C = e^{-(1+\alpha)\tilde{\phi}_w}/\alpha^{3/2}$, $D = e^{-\tilde{\phi}_w} \sqrt{\phi_w/\pi(1+\alpha)/\alpha}$, $G = \alpha[3/2 + (1+\alpha)\tilde{\phi}_w]$.

In equations (17) and (18) the non-local dependence of the source and wall parameters on local quantities is explicit via $\Gamma_s$, $T_s$, $\alpha$, and most prominently $\phi_w$, which influences multiple terms. The explicit non-local dependence in equations (17) and (18) echoes the observation of Hazeltine, in that the influence of the source and wall should enter explicitly in establishing a clear non-local description the parallel heat flux. It is exactly this non-local physics that warrants this closure model being chosen for this study, as it provides a useful benchmark limit, due to the presence of crucial non-local physics via source and boundary influence, for a learned neural network surrogate model.
III. DETAILS OF THE LEARNING METHOD

In this section, we outline a workflow for machine learning the three types of closure introduced previously. The focus will be on bypassing the analytical expressions using neural network formulations. Individual networks for each closure will be designed according to the physical assumptions utilized in their analytical counterpart. In this way, we aim to provide a framework for generating closures in a physics-informed manner that simplifies the network designer’s task for synthesizing closure surrogates. We note that the ultimate goal of this research is to generate closures that can deploy differing underlying assumptions according to localized deployment requirements.

A. A surrogate Braginskii closure

The underlying assumptions of the Braginskii closure rely on the specification of an analytical expression which is pointwise in nature. To that end, our problem is formulated in a supervised learning framework wherein the deep neural network is tasked with predicting the Braginskii closure at discrete locations on a grid locally. Training data is collected by utilizing the analytical form of the closure and framed in a feature-target representation where the features (or inputs) are given by the density \( n(x) \), velocity \( u(x) \), gradients of velocity \( \nabla_x u(x) \), and temperature \( T(x) \) at a given point. The targets are given by the upper-diagonal components of \( \Pi_{BG}^0(n, u, T)(x) \). Note that the diagonal components are all zero and the target tensor is symmetric.

Training and testing data for the Braginskii closure surrogate was generated via equation (8) from randomly generated profiles for \( (n, u, T) \). These profiles were specified to be continuous, smooth, and physical (i.e. \( n, T > 0 \)) via random mode sinusoids

\[
\begin{align*}
n(x, y, z) &= 0.5 \sin(\alpha_1 x) \sin(\alpha_2 y) \sin(\alpha_3 z) + 0.5, \\
u_x(x, y, z) &= \sin(\alpha_4 x), \\
u_y(x, y, z) &= \cos(\alpha_5 y), \\
u_z(x, y, z) &= \sin(\alpha_6 z), \\
T(x, y, z) &= 0.5 \cos(\alpha_7 x) \sin(\alpha_8 y) \cos(\alpha_9 z) + 0.5,
\end{align*}
\]

where random modes \( 0 \leq \alpha_i \leq 3 \) are generated from a uniform distribution.
B. A surrogate Hammett-Perkins closure

The collisionless assumption of the Hammett-Perkins closure necessitates global context for pointwise predictions. This is reflected in the analytical expression defined in Fourier space with its global support. To incorporate this global context into a machine learning framework, we utilize a fully connected neural network that has global support. In other words, a fully connected multi-layered perceptron is used to go from the inputs - given by the temperature profile to the targets - given by the heat-flux. The training data set for this assessment is generated from a temperature profile $\tilde{T}$ that has a constant mean of 1.0 with a global context parameterized by a covariance given by

$$
\tilde{T}(x, x') = \sigma^2 \exp \left( \frac{-2 \sin(\pi(x - x')/T_p)^2}{L_c^2} \right)
$$

(24)

where $T_p$ is the periodicity of the samples and $L_c$ is the correlation length. Sampling from this multivariate distribution sequentially forms the inputs for our training data.

The mechanism for generating the training data also allowed for studying the effect of extrapolation by generating data from different covariance kernels. This will be demonstrated through visualization and quantitative assessments in the following sections.

C. A surrogate Guo-Tang closure

Training and testing data for the Guo-Tang closure is generated by evaluating non-local closure variables in equations (17) and (18) for a variety of randomly generated magnetic field and ambipolar potential profiles. Physical constraints require these profiles to be monotonic and decreasing. To achieve profiles that are widely varying, to provide good training data, but also monotonic decreasing we use the exponential of a Gaussian cumulative distribution function, with random mean and variance. Over a domain $0 \leq x \leq 100$ we generate input profiles

$$
B(x) = B_s \exp \left( -\frac{1}{2} \left[ 1 + \text{erf} \left( \frac{x - \hat{X}_1}{\sqrt{2} \sigma_1} \right) \right] \right),
$$

(25)

$$
\phi(x) = \exp \left( -\frac{1}{2} \left[ 1 + \text{erf} \left( \frac{x - \hat{X}_2}{\sqrt{2} \sigma_2} \right) \right] \right),
$$

(26)

where $B_s = 2$, Gaussian means $10 \leq \hat{X}_i \leq 90$ and variances $10 \leq \sigma_i^2 \leq 60$ are randomly generated from a uniform distribution. Further, the temperature of source electrons $0.5 \leq T_s \leq 2$ is randomly generated from a uniform distribution.
IV. MACHINE LEARNING ARCHITECTURES

For the purpose of building surrogates for each closure hypothesis, we test three different types of machine learning frameworks given by fully connected, convolutional and locally connected artificial neural networks. A brief introduction is provided to all three frameworks and we demonstrate that each framework is associated with certain advantages (as well as disadvantages) and outline the need for surrogate closure development from data based on the underlying hypotheses in a physics-informed manner. In addition, we also comment on the choice of optimal hyperparameters for each of these frameworks and how these may be tied to the nature of non-locality in closure requirements.

A. Fully connected neural network

A fully connected neural network consists of a series of global operations that nonlinearly transform a set of inputs to a set of outputs given by input and output training data. In essence, all the inputs, which in our case are given by pointwise quantities, are flattened into a long vector of inputs and our outputs (also pointwise) are arranged in a similar fashion to have a point to point correlation between these vectors. Multiple fields correspond to multiple samples and therefore several fields of training data are needed to optimally train the free parameters of these networks. The nonlinear transformations in a fully connected network or otherwise are applied by multiple matrix operations on the flattened vector (which may or may not be used for transforming the size of the input vector) followed by subsequent nonlinear transformations of the transformed vector by activations which are element-wise operations of nonlinear functions such as the sigmoid, hyperbolic tangent or rectified linear activations. An example of a sigmoidal activation is

$$\varphi(\beta) = \left(1 + e^{-\beta}\right)^{-1},$$

(27)

where $\beta$ is a component of a vector. Similarly the hyperbolic tangent (also known as tanh) activation is given by

$$\varphi(\beta) = \left(e^{\beta} - e^{-\beta}\right) \cdot \left(e^{\beta} + e^{-\beta}\right)^{-1},$$

(28)

and the rectified linear activation (also denote ReLU) is given by

$$\varphi(\beta) = \max[0, \beta].$$

(29)
We note that ReLU is commonly used for its success in very deep networks as it prevents the saturation inherent in negative exponentiation in the sigmoid and tanh activations.

B. Convolutional neural network

Convolutional neural networks (CNNs) are a class of deep, feed-forward artificial neural networks that are commonly applied to analyzing images. CNNs assume that the underlying images (and in our case field quantities) are stationary (i.e., statistics of one part of the image are the same as other) and that a spatially local correlation exists between the image pixels. This is particularly important for modeling the artifacts introduced by lossy compression algorithms, whose effect on the underlying image is not dependent on the spatial location of the underlying image features. The convolutional layer is the core building block of a CNN and consists of filters whose size defines the extent of spatial locality assumed; each filter corresponds to a specific feature or pattern in the image. The convolution operation using these filters in each layer ensures stationarity and thus translational invariance. Several convolutional layers are stacked in such a way that the complex image features are learned hierarchically by composing together the features in previous layers. The CNN architectures are ideal for analyzing images, are more efficient to implement than fully connected models, and vastly reduce the number of parameters in the network, thus decreases the memory requirement and training time. However, it is generally observed that a greater amount of training data is needed for learning, particularly for spatial maps that need to be deployed point-wise. Greater details may be found in the seminal work of\textsuperscript{22} which led to mainstream utilization of CNNs for image type problems.

C. Locally connected neural network

A locally connected neural network is a hybrid between CNN and FCNN architectures. It leverages non-local relationships in its input layer through feature engineering before using a fully connected network to map to the target values. More importantly, in the process of feature engineering, each grid point in the field is considered a sample. In this way, relationships are learnt pointwise without passing entire fields through the framework of large network architectures. Far simpler neural networks are obtained and training can be performed with fewer generation of fields since the number of samples is dramatically increased. The reader is directed to an example of
such networks in [23] for grid based learning problems. Note that for physics which are inherently
global, this approach is limited as shall be demonstrated for the Hammett-Perkins closure.

V. RESULTS

In this section we utilize statistical assessments for diagnosing the accuracy of machine learned
predictions for the different types of architectures. These assessments are made through plots
showing the probability distribution of machine learning responses compared to those of the truth.
We also show scatter plots comparing predictions and truth. Each type of closure is assessed with
the three types of neural network formulations we have outlined in the previous section both in
terms of accuracy of predictions as well as the number of training parameters and the cost of
training. We then make conclusions about how to select a particular architecture given a priori
understanding of the non-locality of a closure modeling scenario. All the networks assessed in the
experiments of this section utilize a learning rate of 0.001 with a training duration determined by
an early stopping criteria of 10. In other words, a section of the training data would be held out
and not used for optimization (i.e., for validation) and training would be terminated if errors on
this set were observed to be higher than the previous best error for more than 10 epochs. Batch
sizes were varied for optimal throughput for all the experiments although results were observed to
be relatively robust to different choices.

A. Learning Braginskii

We outline the results from using the machine learning model for predicting the Braginskii
stress tensor in this subsection. Here, 6000 data points were utilized randomly from ten fields of
$N_x \times N_y \times N_z = 50 \times 50 \times 50$ for the input and output quantities. We note that the predictions were
for the off-diagonal components of a symmetric tensor (i.e., there were three only outputs from
each query of the network). We shall provide three sets of results for the testing data (coming from
a completely different field of inputs and outputs). Therefore, these plots describe the learning
prowess of the networks on the data that it has not seen during trainable parameter optimization.

Figure 2 shows the probability density functions for the true and predicted values of the testing
data set when using the locally connected neural network. This network architecture utilized each
data point on the grid as a training sample and therefore built a map from 14 inputs (corresponding
to the density, velocity, velocity gradient, and temperature information available from the grid) to 3 outputs corresponding to the closure terms. Response trends for each component are reproduced appropriately by this version of the surrogate formulation. Figure 3 shows the corresponding scatter plots for the same predictions. The 45 degree line represents the true values plotted against themselves. The predictions are seen to be quite close to the true values, represented by the predicted scatter points being close to the true line. The locally connected network utilized solely 1 hidden layer and with 40 neurons in each layer. We note that the coefficient of determination of this experiment was around $R^2 = 0.99$ indicating a very good fit.

Figure 2. Probability density functions of upper diagonal Braginskii stress tensor elements (a) $\tau_{12}$, (b) $\tau_{13}$, (c) $\tau_{23}$ showing true and values predicted by the trained locally connected machine learning framework. Note that this data is a part of testing set showing the generalizability of the trained network.

Next we assess the ability of a CNN for the same prediction task. It is common knowledge that CNNs require more training data to learn effectively and we generate 100 fields of the input and target variables, representing a data set that is ten times larger than that used for the locally connected neural network. Training duration was also much reduced due to the greater number of floating point operations relative to one sample. We remind the reader that a data point for the CNN is an entire field. The results for learning the Braginskii closure using a CNN are shown in terms of PDFs in Figure 4 and scatter plots in Figure 5 which show that learning may be improved (if compared to the locally connected network). We hypothesize that a greater amount of training data may improve the learning of the framework. The hyperparameters of the CNN for this assessment are given by 6 convolution layers with a filter sequence of $[14,30,25,20,15,10,3]$. The filter sequence controls the dimensionality of the number of channels of the data as it is being transformed through the network. To elaborate, the input data has 14 channels corresponding to all
Figure 3. Scatter plots of upper diagonal Braginskii stress tensor elements (a) $\tau_{12}$, (b) $\tau_{13}$, (c) $\tau_{23}$ describing the accuracy of the locally connected machine learning framework. Deviations from the 45 degree line indicate erroneous predictions. Note that this data is a part of testing set showing the generalizability of the trained network.

The input quantities available on the grid and the output tensor contains 3 channels that correspond to the closure terms. No pooling layers were used in this study and zero padding was used to preserve the dimension of the field during the forward pass through the framework. The $R^2$ of this particular experiment was slightly lower, but still very high, at 0.95.

Figure 4. Probability density functions of upper diagonal Braginskii stress tensor elements (a) $\tau_{12}$, (b) $\tau_{13}$, (c) $\tau_{23}$ showing true and values predicted by the trained convolutional neural network learning framework. Note that this data is a part of testing set showing the generalizability of the trained network.

Finally we show results from a fully connected network which interprets every field quantity at every points to a one dimensional vector and maps to the target tensor components also interpreted to be part of a large vector. Basically, our fully connected map is given by a nonlinear
transformation from a space of dimension $N_x \times N_y \times N_z \times 14$ to $N_x \times N_y \times N_z \times 3$. One can note immediately, that the use of a fully connected network that connects every point in the field is computationally infeasible. Here, we demonstrate that training such a framework is also non-trivial as shown in Figures 6 (for the PDF) and Figures 7. $R^2$ values of approximately 0.1 indicated the poor performance of this approach.

Figure 6. Probability density functions of upper diagonal Braginskii stress tensor elements (a) $\tau_{12}$, (b) $\tau_{13}$, (c) $\tau_{23}$ showing true and values predicted by the trained fully connected neural network learning framework. Note that this data is a part of testing set showing the generalizability of the trained network.
Figure 7. Scatter plots of upper diagonal Braginskii stress tensor elements (a) $\tau_{12}$, (b) $\tau_{13}$, (c) $\tau_{23}$ describing the accuracy of the fully connected neural network framework. Deviations from the 45 degree line indicate erroneous predictions. Note that this data is a part of testing set showing the generalizability of the trained network.

B. Learning Guo-Tang

We proceed by assessing the ability of the machine learning frameworks for predicting the heat flux in the Guo-Tang framework. Figure 8 shows the probability density functions of the true and predicted values of the parallel and perpendicular flux for the locally connected neural network. In addition they also show scatter plots and a sample realization of the true and predicted closure profile for one prediction. The locally connected neural network is able to obtain the right trends in the parallel and perpendicular fluxes from the point of view of PDFs and the scatter plots show a good agreement as well. Our locally connected neural network utilizes 4 hidden layers with 50 neurons in each layer to construct the local map between grid variables given by an input space of dimension 3, given by the magnetic field, ambipolar potential and source electron temperature, and output space of dimension 2, given by the two parallel and perpendicular components of the heat flux closure. We note that our grid is of dimension 256 but data at each location of the grid is considered independent of its neighbors for the purposes of training and deployment. $R^2$ values of 0.98 were observed for this experiment.

Figure 9 shows results from a deployment of the CNN framework for the same task. Similar trends are obtained for this assessment as well. However, we note that some boundary inaccuracies can be observed due to the strided nature of convolutional neural networks. These assessments utilized a zero-padding at the boundaries but more involved boundary conditions can also be em-
bedded such as a periodic padding for each convolutional filter. Overall, the CNN is also able to learn the right nonlinear relationship for this closure. Our CNN architecture uses 6 convolutional layers with a filter sequence of \([2,30,25,20,15,10,3]\) and converged to an \(R^2\) value of 0.98.

Figure 10 shows results from a deployment of the fully-connected neural network. Good results are obtained for this assessment although field values can be observed to be slightly noisy. This noise is a result of the fully connected network being sensitive to perturbations across the entire domain, hence small fluctuations of input variables through the domain can lead to jitter in the observed output quantities through the entire domain as well. However, despite the presence of noise, the predictions from the fully connected framework show less deviation (particularly in the case of the parallel flux) from the true values. This might suggest a suitable use of the framework along with a low-pass spatial kernel used for postprocessing the output. We note that our neural network utilizes 4 hidden layers with 50 neurons in each layer to construct the local map between grid variables given by an input space of dimension 256x3 and output space of dimension 256x2 with converged \(R^2\) values of 0.98.

C. Learning Hammett-Perkins

We perform a third analysis (similar to the previous two) for the Hammett-Perkins closure with assessments for the locally connected neural network shown in Figure 11. The global nature of Hammett-Perkins is immediately discernible through the performance of the trained local framework which fails to capture the right trends of the output almost entirely. This was observed for multiple different choices of the hyperparameters of the network. The results here are shown for 2 hidden layers and 30 neurons in each layer and attempt to map from a 1 dimensional input of temperature profile to a 1 dimensional output of the heat flux.

In contrast, Figure 13 shows the results of a CNN with a finite non-locality where it is observed that some trends are learnt although the accuracy is marginal. The size of the kernel and the number of CNN layers are instrumental in the accuracy of the predictions. Essentially, an increase in the number of layers with a local stencil increases global influence and may be assumed to improve accuracy. However, we note that the perfectly global nature of Hammett-Perkins implies that nothing short of a fully connected network can obtain optimal learning. These results are shown for a filter sequence of \([1,30,25,20,15,10,1]\) with only one input and one output and obtained \(R^2\) values of 0.94.
Table I. The number of trainable parameters for each neural network architecture for the three different closure scenarios. It is apparent that the fully connected neural networks require a large number of parameters in comparison to the CNN or locally connected networks.

| Closure type       | Fully connected | Convolutional | Locally connected |
|--------------------|-----------------|---------------|-------------------|
| Braginskii         | 64,128,750      | 58,133        | 723               |
| Guo-Tang           | 72,212          | 5,532         | 7,952             |
| Hammett-Perkins    | 16,576          | 5,321         | 1,021             |

Table II. The mean-squared errors for each neural network architecture for the three different closure scenarios.

| Closure type       | Fully connected | Convolutional | Locally connected |
|--------------------|-----------------|---------------|-------------------|
| Braginskii         | $1.10 \times 10^{-2}$ | $1.66 \times 10^{-4}$ | $2.40 \times 10^{-5}$ |
| Guo-Tang           | $2.93 \times 10^{-4}$ | $1.09 \times 10^{-3}$ | $4.84 \times 10^{-4}$ |
| Hammett-Perkins    | $7.54 \times 10^{-4}$ | $2.71 \times 10^{-2}$ | $1.97 \times 10^{-1}$ |

The results of the fully connected network are shown in Figure 13 where a much improved performance is obtained by the machine learning architecture. The fully connected neural network mimics the global nature of the Fourier-space transformation and is thus able to recover the trends most accurately. A conclusion from this set of experiments is that for prediction tasks where the interaction between inputs is perfectly global, a fully connected network is optimally suited. However, due to the greater computational expense in training and deployment of these networks, one may also utilize a sufficiently deep CNN. The fully connected networks were able to obtain large $R^2$ values of 0.99 and are optimally suited for this type of closure requirement. The architecture for this task was given by 2 hidden layers with 30 neurons mapping inputs from 256 dimensional inputs to 256 dimensional outputs.

To summarize the differing complexities of the proposed network architectures, we outline the number of trainable parameters for each closure learning in Table I.
D. A note on extrapolation

In the previous sections, we have demonstrated the applicability of learned surrogates when subject to the domain of the training data. In practice, one may have a lot of data to learn from but that data may not have a broad span of a system’s parameter space to generate training data; this may be considered as a balance between big data, i.e. a large quantity, and broad data, a large and statistically meaningful quantity of data. Thus, predictions may have to be extrapolated based on what was learnt from the available training domain. Before presenting outcome of the closure surrogates in extrapolating regimes, we wish to note that extrapolating outside the training domain using a learned neural network architecture is a non-trivial task that requires careful application. A simple demonstrative example of the unreliability of trying to emulate even the most simple of functions, the identity \( f(x) = x \), was presented recently\(^{24}\). While an exact analytic solution of network coefficients can be found for a simple network architecture like the identity function, to allow accurate extrapolation outside a training domain, Gin et al. highlight the sensitivity of learned network coefficients and thus the reliability of predictions outside the training domain even for such a simple problem. We believe this is an important lesson to keep in mind when applying neural network methods to real problems, and emphasizes that a network may learn to mimic behaviour of a set of training data but cannot reliably learn the underlying general functional relationships relating input and output quantities.

E. Testing for extrapolation

One of the key challenges of machine learning is to detect extrapolation. Machine learning frameworks are notoriously unreliable in extrapolating regimes as mentioned previously. Detecting extrapolation is useful for determining if more data should be generated and the framework should be retrained using this new data before deployment. For example, we outline the performance of the fully connected neural network used to train the Hammett-Perkins closure in the previous section for a data point which is generated from a slightly different underlying distribution and the result is shown in Figure 14. It can be seen that the true profile has more high frequency content which the trained network struggles to predict even though low frequency trends are recovered somewhat. The PDF of the true dataset is also recovered approximately (note how this new data set is distinctly multi-modal). The scatter plots also show consistent errors through the
entirety of the domain.

Finally, we outline the use of T-distributed Stochastic Neighbor Embedding (t-SNE) which is a nonlinear embedding of our training and testing data onto a two-dimensional space. The embedding only requires the input data (i.e., the true values need not be known) and can be a check to determine the risk of extrapolation. Figure 15 shows this analysis on the new dataset with the multimodal distribution of outputs which leads to the presence of two distinct clusters corresponding to the two distinct distributions in our total dataset. Note however, that this method cannot be coupled with a computational deployment of a closure as it is significantly costly in comparison with a forward pass through any of these networks. However, it can be a useful diagnostic to determine if a machine learning framework has contributed to failure of a simulation.

VI. CONCLUSION

From this study, we find that in different physical regimes, an informed choice can be made on the neural network architecture chosen to formulate the closure surrogate model based on the physics expected in the model system. A simple schematic summary of this is shown Fig. 16, demonstrating that for a given knowledge of the locality of the system, one can choose a network architecture that is suitable to the physics inherent in the problem.

In the simplest limit is local closure, where point-wise values of closed quantities can be expressed as a functions of macroscopic values at that exact location in space. The primary example of this regime used in this work was the Braginskii closure for a collisional magnetized plasma, in which equilibration through collisions enables a local closure to be formulated for the variables that require closure. In this scenario, we observe that a simple single layer network can robustly and efficiently reproduce the required physics in high dimensional simulations without excessive computational burden that a more general connected network architecture would produce.

Next, one can also consider intermediate regimes where non-locality exists but is contained over a finite range of influence. This concept has previously been identified by Hazeltine in the context of source and boundary edge properties influencing the dynamics of the bulk plasma. In this scenario, we observe a CNN architecture, similar to that commonly used in image processing where there is a relationship between neighbouring points, yields a good balance between reproducing the required non-local physics, without a fully connected multilayered network.

Finally, in the extreme of a globally non-local closure, such as in the case in which collisionless-
phase-mixing provides a truly global influence on point-wise values, we find that a fully connected multilayered deep neural network is required to emulate the global non-locality of the closure.

ACKNOWLEDGMENTS

We acknowledge productive discussions with Dr. Sandeep Madireddy and Dr. Bethany Lusch for this article. This material is based upon work supported by the U.S. Department of Energy (DOE), Office of Science, Office of Advanced Scientific Computing Research, under Contract DE-AC02-06CH11357. This research was funded in part and used resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357. Part of this work was performed at Los Alamos National Laboratory (contract No. 89233218CNA000001) and jointly supported by the Theory Program of the Office of Fusion Energy Sciences, and two SciDAC projects on Tokamak Disruption Simulation (TDS) and runaway electrons (SCREAM) by Office of Fusion Energy Science and Office of Advanced Scientific Computing.

REFERENCES

1. F. Chen, *Introduction to plasma physics and controlled fusion. Third edition.* (Springer International Publishing, 2016).
2. S. I. Braginskii, “Transport Processes in a Plasma,” *Reviews of Plasma Physics* 1, 205 (1965).
3. P. Helander and D. Sigmar, *Collisional Transport in Magnetized Plasmas*, Cambridge Monographs on Plasma Physics (Cambridge University Press, 2005).
4. E. D. Held, J. D. Callen, C. C. Hegna, C. R. Sovinec, T. A. Gianakon, and S. E. Kruger, “Nonlocal closures for plasma fluid simulations,” *Physics of Plasmas* 11, 2419–2426 (2004), [https://doi.org/10.1063/1.1645520](https://doi.org/10.1063/1.1645520).
5. S. Yarlanki, B. Rajendran, and H. Hamann, “Estimation of turbulence closure coefficients for data centers using machine learning algorithms,” in *13th InterSociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems* (2012) pp. 38–42.
6. C.-W. Chang and N. T. Dinh, “Classification of machine learning frameworks for data-driven thermal fluid models,” *International Journal of Thermal Sciences* 135, 559 – 579 (2019).
7. O. San and R. Maulik, “Machine learning closures for model order reduction of thermal fluids,” Applied Mathematical Modelling 60, 681–710 (2018).
8. K. Montes, C. Rea, R. Granetz, R. Tinguely, N. Eidietis, O. Meneghini, D. Chen, B. Shen, B. Xiao, K. Erickson, and M. Boyer, “Machine learning for disruption warnings on alcator c-mod, DIII-d, and EAST,” Nuclear Fusion 59, 096015 (2019).
9. “Predicting disruptive instabilities in controlled fusion plasmas through deep learning,” Nature 568, 526–531 (2019).
10. D. Wroblewski, G. Jahns, and J. Leuer, “Tokamak disruption alarm based on a neural network model of the high-beta limit,” Nuclear Fusion 37, 725–741 (1997).
11. D. Rastovic, “Targeting and synchronization at tokamak with recurrent artificial neural networks,” Neural Computing and Applications 21, 1065–1069 (2012).
12. E. Coccorese, R. Martone, and F. C. Morabito, “A neural network approach for the solution of electric and magnetic inverse problems,” IEEE Transactions on Magnetics 30, 2829–2839 (1994).
13. C. Ma, B. Zhu, and X. Xu, “Machine learning surrogate models for landau fluid closure,” (2019).
14. G. W. Hammett and F. W. Perkins, “Fluid moment models for Landau damping with application to the ion-temperature-gradient instability,” Phys. Rev. Lett. 64, 3019–3022 (1990).
15. Z. Guo and X.-Z. Tang, “Parallel Heat Flux from Low to High Parallel Temperature along a Magnetic Field Line,” Phys. Rev. Lett. 108, 165005 (2012).
16. M. Raissi, P. Perdikaris, and G. Karniadakis, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” Journal of Computational Physics 378, 686–707 (2019).
17. J.-L. Wu, H. Xiao, and E. Paterson, “Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework,” Phys. Rev. Fluids 3, 074602 (2018).
18. J. Bedrossian and N. Masmoudi, “Inviscid damping and the asymptotic stability of planar shear flows in the 2d euler equations,” (2013).
19. P. J. Morrison, “Hamiltonian description of vlasov dynamics: Action-angle variables for the continuous spectrum,” Transport Theory and Statistical Physics 29, 397–414 (2000), https://doi.org/10.1080/00411450008205881.
20. J. M. Heninger and P. J. Morrison, “An integral transform technique for kinetic systems with collisions,” Phys. Plasmas 25, 082118 (2018).
21. R. D. Hazeltine, “Transport theory in the collisionless limit,” Physics of Plasmas 5, 3282–3286 (1998), https://doi.org/10.1063/1.872996.

22. A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in neural information processing systems (2012) pp. 1097–1105.

23. R. Maulik and O. San, “A neural network approach for the blind deconvolution of turbulent flows,” Journal of Fluid Mechanics 831, 151–181 (2017).

24. C. Gin, B. Lusch, S. L. Brunton, and J. N. Kutz, “Deep learning models for global coordinate transformations that linearize pdes,” arXiv preprint arXiv:1911.02710 (2019).

25. G. E. Hinton and S. T. Roweis, “Stochastic neighbor embedding,” in Advances in neural information processing systems (2003) pp. 857–864.

26. G. W. Hammett, W. Dorland, and F. W. Perkins, “Fluid models of phase mixing, landau damping, and nonlinear gyrokinetic dynamics,” Physics of Fluids B: Plasma Physics 4, 2052–2061 (1992), https://doi.org/10.1063/1.860014.

27. D. Tsiklauri and T. Haruki, “Physics of collisionless phase mixing,” Physics of Plasmas 15, 112902 (2008) https://doi.org/10.1063/1.3023157.
Figure 8. Machine learning results for learning the Guo-Tang closure for the locally connected machine learning framework. The top row shows PDFs of the true and predicted fluxes, the middle row shows the scatter plots for the same and the bottom row shows a sample field prediction by the trained frameworks. All assessments are on the test data set.
Figure 9. Machine learning results for learning the Guo-Tang closure for the convolutional neural network learning framework. The top row shows PDFs of the true and predicted fluxes, the middle row shows the scatter plots for the same and the bottom row shows a sample field prediction by the trained frameworks. All assessments are on the test data set.
Figure 10. Machine learning results for learning the Guo-Tang closure for the fully connected machine learning framework. The top row shows PDFs of the true and predicted fluxes, the middle row shows the scatter plots for the same and the bottom row shows a sample field prediction by the trained frameworks. All assessments are on the test data set.
Figure 11. Results from machine learning the Hammett-Perkins closure with probability density functions (left), scatter plots (middle) and a sample prediction visualized against the truth (right) for the locally connected neural network. These are generated for the testing data set.

Figure 12. Results from machine learning the Hammett-Perkins closure with probability density functions (left), scatter plots (middle) and a sample prediction visualized against the truth (right) for the convolutional neural network. These are generated for the testing data set.
Figure 13. Results from machine learning the Hammett-Perkins closure with probability density functions (left), scatter plots (middle) and a sample prediction visualized against the truth (right) for the fully connected neural network. These are generated for the testing data set.

Figure 14. Results from machine learning the Hammett-Perkins closure with probability density functions (left), scatter plots (middle) and a sample prediction visualized against the truth (right) for a fully connected neural network. These are generated for the testing data set from a different underlying distribution. Results of extrapolation outside the space of trained input data can be clearly observed.
Figure 15. T-SNE encoding of training and testing datasets from different underlying distributions. The two clusters indicate two distinct distributions between training and testing. Formation of distinct clusters in input data is a key indicator of operating in an extrapolating fashion, in which care must be taken.

Figure 16. Simple schematic of varying classes of closure formulations. In each figure a red focus point can be thought to have a non-local interaction length, $\lambda_{NL}$, bounded by the green region containing blue points that influence the dynamics of a given focus point. This interaction region can be qualified in each region as (a) $\lambda_{NL} \to 0$ and local information is sufficient, (b) finite $\lambda_{NL}$ and finite region of interaction, and (c) $\lambda_{NL} \to \infty$ and the entire global domain must be considered.