Artificial neural network subgrid models of 2D compressible magnetohydrodynamic turbulence

Shawn G. Rosofsky[1,2] and E. A. Huerta[1,3]

[1]*NCSA, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA*
[2]*Department of Physics, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA*
[3]*Department of Astronomy, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA*

(Received 23 December 2019; accepted 17 March 2020; published 9 April 2020)

We explore the suitability of deep learning to capture the physics of subgrid-scale ideal magnetohydrodynamics turbulence of 2D simulations of the magnetized Kelvin-Helmholtz instability. We produce simulations at different resolutions to systematically quantify the performance of neural network models to reproduce the physics of these complex simulations. We compare the performance of our neural networks with gradient models, which are extensively used in the magnetohydrodynamic literature. Our findings indicate that neural networks significantly outperform gradient models in accurately computing the subgrid-scale tensors that encode the effects of magnetohydrodynamics turbulence. To the best of our knowledge, this is the first exploratory study on the use of deep learning to learn and reproduce the physics of magnetohydrodynamics turbulence.

DOI: 10.1103/PhysRevD.101.084024

I. INTRODUCTION

In astrophysical simulations of magnetohydrodynamics (MHD) such as magnetized binary neutron star (BNS) mergers, we confront turbulent phenomena in the limit of an infinite Reynolds number Re [1]. While these turbulent effects are often ignored, ultrahigh-resolution simulations of magnetized BNS mergers have demonstrated that MHD turbulence can amplify the magnetic field by several orders of magnitude and occur only at resolutions that are too computationally expensive to run in bulk [2,3]. This amplification is due to the magnetized Kelvin-Helmholtz instability (KHI), which occurs when two fluids flow past each other in opposite directions. In order to help resolve turbulence originating from the KHI, we examine the methods employed in more traditional hydrodynamical turbulence.

To resolve turbulent effects, the computational fluid dynamics community uses several classes of simulations that provide varying degrees of accuracy. Direct numerical simulations (DNSs) provide the most accurate results by capturing all the effects at all scales relevant to the problem being studied. To resolve the turbulent effects of these simulations, DNSs require extremely high resolutions that scale as the cube of Re. This resolution requirement renders DNSs feasible only for a small number of simulations. Moreover, our problems of interest has an extremely high Re, resulting in DNSs becoming too computationally expensive for our work.

The other prominent techniques, Reynolds-averaged Navier-Stokes (RANS) and large eddy simulations (LESs), employ subgrid-scale (SGS) models to reproduce the most important effects of DNSs such as the energy transfer rate at much lower resolutions. RANS is the most widely used but is best suited for steady state phenomena. For an instability such as the KHI, LESs serve as the preferred approach. The goal behind LESs is to evolve the equations with sufficient resolution to resolve the largest eddies and rely on the SGS model to compute the contribution of the smaller eddies.

Recent work has sought to develop SGS models of MHD turbulence using traditional LES models [4–15]. However, MHD turbulence presents some unique challenges not observed in standard hydrodynamical turbulence. Although the boundary conditions for problems of interest are typically much simpler, the equations are more complex. These complexities include a dynamo mechanism for the conversion between kinetic and magnetic energy as well as anisotropies arising from the magnetic field [15–17]. Moreover, there exists a much weaker understanding of MHD turbulence compared to the hydrodynamical variety.

To resolve these complexities without exerting significant efforts studying the intricacies of MHD turbulence, we explore the use of artificial neural networks (ANNs) to act as SGS models. Significant work has been done in examining and evaluating ANN models of hydrodynamical turbulence for both RANS and LESs in recent years [18–26]. These studies indicate that ANNs may outperform traditional approaches used to model turbulence.

In this article, we develop a proof-of-concept neural network model to quantify the performance of deep-learning algorithms to reproduce the true dynamics of

turbulent magnetic field amplification at manageable resolutions of MHD simulations of the KHI in the LES formalism. We use as a driver for this study 2D MHD simulations and compare the performance of our neural networks to traditional models, such as the *a priori* study introduced in Ref. [13]. For reference, an *a priori* study involves evaluating the performance of the models in how closely they reproduce the SGS effects compared to the filtered DNS data. In contrast, an *a posteriori* study would implement these models in an actual simulation to observe how the SGS models compare to the higher-resolution DNS simulations. We leave the more computationally expensive 3D case as well as the implementation of these neural network models, and subsequent *a posteriori* comparison of the models, to future work. Herein, we will perform a more in-depth analysis of the conditions each model performs best in our *a priori* study to gain as much insight as possible before moving to the more complicated tests.

This article is organized as follows: Sec. II provides an overview of the LES formalism and its application to the MHD equations. In Sec. III, we describe the SGS models used in this work, including our proposed ANN model and the traditional gradient model. Section IV describes the simulations used to train and evaluate our SGS models. We describe the methods in which those simulations were employed to train the ANN model in Sec. V. In Sec. VI, we define the metrics used to evaluate the SGS models. We provide the results of our *a priori* study of the ANN SGS turbulence model and compare its performance with that of the gradient model. Section VIII summarizes our findings and outline future directions of work.

II. LES FORMALISM

In this section, we introduce the mathematical formalisms that we will use throughout the article. We describe the LES formalism and briefly describe the compressible MHD equations, which will be used as the science driver of our analysis.

A. Filtering

In the LES formalism, one views the grid resolution as a spatial filter applied to a continuous variable. In this approach, the size of the grid Δ corresponds to the size of the filter. Typically, we start with very-high-resolution data taken from DNS or experimental results and apply a filter with a cutoff size Δ_f, where $\Delta_f > \Delta$ is the lower-resolution grid on which we want to perform our simulation. We apply the kernel G to a field f as

$$\bar{f}(\mathbf{x},t) = \int_{-\infty}^{\infty} G(\mathbf{x}-\mathbf{x}')f(\mathbf{x}',t)d\mathbf{x}'. \tag{1}$$

For implicit LES simulations which are employed in this work, the filtering operator of size Δ_f is applied to the high-resolution simulation of grid size Δ when calibrating SGS models. In turn, this filtering provides insight into the effect of moving to a lower to a lower grid resolution. The choice of filter depends on the numerical method employed. For finite volume schemes like those used in this work, a box or top-hat filter is used to simulate the spatial averaging that occurs during such schemes. This filter kernel is given in real space for D spatial dimensions as

$$G(|\mathbf{x}-\mathbf{x}'|) = \prod_{i=1}^{D} G_i(|x_i - x_i'|), \tag{2}$$

where

$$G_i(|x_i - x_i'|) = \begin{cases} 1/\Delta_f & \text{if } |x_i - x_i'| \le \Delta_f/2, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Filtering operators commute with linear terms. However, nonlinearities in the MHD equations fail to commute with the filtering operator. This results in a residual term known as the SGS tensor. We will provide examples of these SGS tensors in the next section.

For compressible fluids, we use a specific type of filtering called Favre or density weighted to simplify our problem by eliminating the SGS tensor in the continuity equation. For some quantity f weighted by some density ρ, we define the Favre filtered quantity $\tilde{f}$ as

$$\tilde{f} = \frac{\bar{\rho f}}{\bar{\rho}}. \tag{4}$$

This also gives us the identity $\overline{\rho f} = \bar{\rho}\,\tilde{f}$.

B. Compressible MHD equations

1. Unfiltered MHD equations

For the evolution of our system, we used the conservative form of the ideal compressible Newtonian MHD equations. Each equation continuity, momentum, induction, and energy evolution, respectively, represents the local evolution of a globally conserved quantity. The equations are given by

$$\partial_t \rho + \partial_i[\rho v^i] = 0, \tag{5}$$

$$\partial_t(\rho v^j) + \partial_i\left[\rho v^i v^j - B^i B^j + \delta^{ij}\left(p + \frac{B^2}{2}\right)\right] = 0, \tag{6}$$

$$\partial_t B^j + \partial_i[v^i B^j - v^j B^i] = 0, \tag{7}$$

$$\partial_t u + \partial_i[(u + p + B^2)v^i - (v_j B^j)B^i] = 0, \tag{8}$$

where the total energy density u is defined as

$$u = e + \frac{\rho v^2}{2} + \frac{B^2}{2}. \tag{9}$$

Here, the indices are spatial components assuming the Einstein summation convention, δ^{ij} is the Kronecker delta, ρ is the mass density, p is the pressure, e is the internal energy density, v^i is the velocity, and B^i is the magnetic field. The units of this expression are such that the speed of light c and the magnetic permeability μ_0 are $c = \mu_0 = 1$. For this system, we used an ideal gas equation of state (EOS) to define p as

$$p = (\gamma - 1)e, \tag{10}$$

where γ is the adiabatic index set to $\gamma = 4/3$ for a relativistic gas in this work. We note that we intentionally did not exploit any simplifications made using the fact that we have an ideal gas EOS to ensure that our ANN turbulence model can be used for any generic EOS. This is done to ensure that the model can be easily employed by BNS simulations where the EOS is a variable parameter.

2. Filtered MHD equations

To derive the filtered equations, we apply Eq. (2) to Eqs. (5)–(8) [13]. We find these equations become

$$\partial_t \bar{\rho} + \partial_i[\bar{\rho}\tilde{v}^i] = 0, \tag{11}$$

$$\partial_t(\bar{\rho}\tilde{v}^j) + \partial_i\left[\bar{\rho}\tilde{v}^i\tilde{v}^j - \bar{B}^i\bar{B}^j + \delta^{ij}\left(\tilde{p} + \frac{\bar{B}^2}{2}\right)\right] = -\partial_i\tau^{ij}_{\rm mom}, \tag{12}$$

$$\partial_t\bar{B}^j + \partial_i[\tilde{v}^i\bar{B}^j - \tilde{v}^j\bar{B}^i] = -\partial_i\tau^{ij}_{\rm ind}, \tag{13}$$

$$\partial_t\bar{u} + \partial_i[(\bar{u} + \tilde{p} + \bar{B}^2)\tilde{v}^i - (\tilde{v}_j\bar{B}^j)\bar{B}^i] = -\partial_i\tau^{i}_{\rm eng} + \Sigma_{\rm eng}, \tag{14}$$

where the merged SGS tensor terms are given by

$$\tau^{ij}_{\rm mom} = \bar{\rho}\tau^{ij}_{\rm kin} - \tau^{ij}_{\rm mag} + \delta^{ij}\left(\frac{1}{2}\delta_{kl}\tau^{kl}_{\rm mag} + (\bar{p} - \tilde{p})\right), \tag{15}$$

$$\tau^{i}_{\rm eng} = \tau^{i}_{\rm enth} + \tau^{ij}_{\rm mom}\tilde{v}_j + \tau^{ij}_{\rm ind}\bar{B}_j, \tag{16}$$

and the scalar SGS tensor terms denoted by Σ are given by

$$\Sigma_{\rm eng} = \Sigma_{\rm pres} + \Sigma_{\rm mom} + \Sigma_{\rm ind}, \tag{17}$$

$$\Sigma_{\rm pres} = \overline{v^i\partial_i p} - \tilde{v}^i\partial_i\tilde{p}, \tag{18}$$

$$\Sigma_{\rm mom} = \frac{1}{2}(\partial_i\tilde{v}_j + \partial_j\tilde{v}_i)\tau^{ij}_{\rm mom}, \tag{19}$$

$$\Sigma_{\rm ind} = \frac{1}{2}(\partial_i\bar{B}_j - \partial_j\bar{B}_i)\tau^{ij}_{\rm ind}. \tag{20}$$

In the above expressions, we have defined

$$\tilde{e} = \bar{u} - \frac{\bar{\rho}\tilde{v}^2}{2} - \frac{\bar{B}^2}{2}, \tag{21}$$

$$\tilde{p} = (\gamma - 1)\tilde{e} \tag{22}$$

and will define the enthalpy h and its filtered version $\tilde{h}$ as

$$h = \rho + e + p, \tag{23}$$

$$\tilde{h} = \bar{\rho} + \tilde{e} + \tilde{p}, \tag{24}$$

respectively.

For modeling the SGS terms in Eqs. (11)–(14), we care only about $\tau_{\rm kin}$ describing turbulent motion, $\tau_{\rm mag}$ describing the contribution of the turbulent magnetic field to the motion, $\tau_{\rm ind}$ describing the turbulent amplification of the magnetic field, and $\tau_{\rm enth}$ describing the effect of turbulence on the energy transfer. We neglect the terms $(\bar{p} - \tilde{p})$ and $\Sigma_{\rm pres}$, as we expect their contributions to be small and EOS dependent, which reduces the robustness of our models. The rest of the terms in Eqs. (11)–(14) are combinations of the aforementioned terms. The four SGS tensors we want to model are defined formally as

$$\tau^{ij}_{\rm kin} = \widetilde{v^iv^j} - \tilde{v}^i\tilde{v}^j, \tag{25}$$

$$\tau^{ij}_{\rm mag} = \overline{B^iB^j} - \bar{B}^i\bar{B}^j, \tag{26}$$

$$\tau^{ij}_{\rm ind} = (\overline{v^iB^j} - \overline{v^jB^i}) - (\tilde{v}^i\bar{B}^j - \tilde{v}^j\bar{B}^i), \tag{27}$$

$$\tau^{i}_{\rm enth} = \overline{hv^i} - \tilde{h}\tilde{v}^i. \tag{28}$$

The astute reader may notice that $\tau^i_{\rm enth}$ is actually a vector, but we will refer to it as an SGS tensor throughout this work for the sake of conciseness.

III. MODELING SGS TENSORS

In this section, we introduce the gradient model, which currently represents the state of the art in the LES MHD literature, and our deep-learning algorithm. In what follows, we will present direct comparisons between these two methodologies to highlight their key differences and to furnish evidence that deep learning outperforms the gradient approach.

A. Gradient model

The gradient model is extensively used in the LES MHD literature [13,15]. The prevalence of this model in other LES MHD turbulence studies promotes it as a good baseline to test the performance of our neural network model. The gradient model is derived using the Taylor

expansion of the SGS stress tensor under a particular filtering operator. Here we use the leading-order expansion of our box filtering operator which is also valid for a Gaussian filter [15] to obtain

$$\overline{fg} \simeq \bar{f}\,\bar{g} + \frac{\Delta_f^2}{12}\partial^i \bar{f}\partial_i \bar{g}, \tag{29}$$

$$\widetilde{fg} \simeq \tilde{f}\,\tilde{g} + \frac{\Delta_f^2}{12}\partial^i \tilde{f}\partial_i \tilde{g}, \tag{30}$$

$$\overline{fg} \simeq \tilde{f}\,\bar{g} + \frac{\Delta_f^2}{12}\partial^i \tilde{f}\left(\partial_i \bar{g} - \frac{\partial_i \bar{\rho}}{\bar{\rho}}\bar{g}\right), \tag{31}$$

for regular filtered terms, Favre filtered terms, and mixed filtered terms, respectively [13]. This results in the following expressions for the SGS tensors [13]:

$$\tau_{\text{kin}}^{ij} = C_{\text{kin}}^{ij}\frac{\Delta_f^2}{12}\partial_k \tilde{v}^i \partial^k \tilde{v}^j, \tag{32}$$

$$\tau_{\text{mag}}^{ij} = C_{\text{mag}}^{ij}\frac{\Delta_f^2}{12}\partial_k \bar{B}^i \partial^k \bar{B}^j, \tag{33}$$

$$\tau_{\text{ind}}^{ij} = C_{\text{ind}}^{ij}\frac{\Delta_f^2}{12}\left[\partial_k \tilde{v}^i\left(\partial^k \bar{B}^j - \frac{\partial^k \bar{\rho}}{\bar{\rho}}\bar{B}^j\right) - \partial_k \tilde{v}^j\left(\partial^k \bar{B}^i - \frac{\partial^k \bar{\rho}}{\bar{\rho}}\bar{B}^i\right)\right], \tag{34}$$

$$\tau_{\text{enth}}^{i} = C_{\text{enth}}^{i}\frac{\Delta_f^2}{12}\frac{\gamma}{\gamma-1}\left[\partial_j \tilde{p} - \tilde{p}\frac{\partial_j \bar{\rho}}{\bar{\rho}}\right]\partial^j \tilde{v}^i. \tag{35}$$

The coefficient C^{ij} is determined by the best fit of the data to a time slice of filtered DNS data for each component of τ_{grad}^{ij} independently. The fitting is determined by

$$C^{ij} = \frac{\sum_{\mathbf{x}_f}(\tau_{\text{DNS}}^{ij}(\mathbf{x}_f)\tau_{\text{grad}}^{ij}(\mathbf{x}_f))}{\sum_{\mathbf{x}_f}\tau_{\text{grad}}^{ij}(\mathbf{x}_f)}, \tag{36}$$

where τ_{grad}^{ij} is the SGS tensor calculated by the gradient model in Eqs. (32)–(35), τ_{DNS}^{ij} is the true SGS tensor computed directly from the DNS data, $\mathbf{x}_f$ represents the filtered grid, and Einstein summation notation is not used. When employing this model in an *a posteriori* test, one would estimate C^{ij} with a secondary filter [22,27]. In the LES literature, this is known as a dynamical model. However, we do not use a secondary filter for our *a priori* study and instead filter the DNS data directly. We acknowledge that this may overestimate the performance of the gradient model compared to an *a posteriori* study.

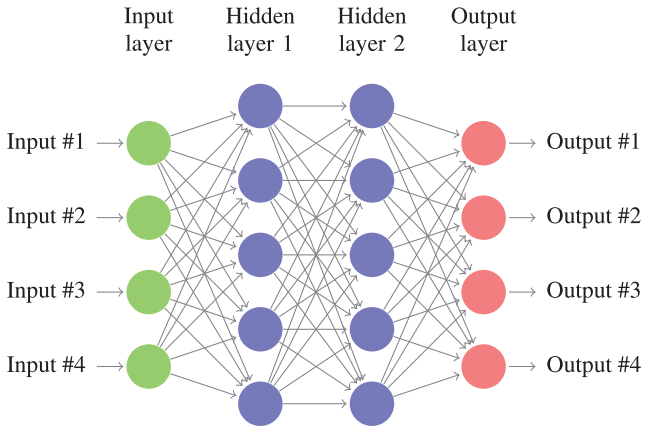

FIG. 1. Schematic illustration of a neural network. A multilayer perceptron with two hidden layers is presented. Circles represent neurons, whereas arrows correspond to weights.

B. Neural network model

ANNs are the building blocks of deep neural networks. The basic units of calculation in ANNs are called neurons, which are connected via weighted inputs that resemble synapses. These biologically inspired models have the proven capability of learning from data, which has accelerated the data-driven discovery revolution over the past decade [28–32].

As shown in Fig. 1, a neural network creates a relationship between the inputs and outputs. This relation uses multiple layers of neurons connected through a series of linear or nonlinear functions. The input layer takes the input data and applies these operations to calculate its outputs X_i^1 for each input i. Then, each of the ANN's subsequent layers l takes the outputs of the previous layer X_j^{l-1} of layer $l-1$ and applies this same calculations to calculate the outputs X_i^l of each of is neurons. The calculation is performed as

$$X_i^l = g(s_i^l + b_i^l), \tag{37}$$

$$s_i^l = \sum_j W_{ij}^l X_j^{l-1}, \tag{38}$$

where g is a nonlinear function known as an activation function, and the parameters to be tuned during training are the weights W_{ij}^l and biases b_i^l. The values of W_{ij}^l and b_i^l are continually adjusted during the training stage until training data with the same labels consistently yield similar results in the output layer X_i^L. In our case, the output of the neural network model corresponds to the SGS tensor components. For the activation function of the hidden layers, we selected the rectified linear unit (ReLU), which is common in machine learning for its fast training speed. The ReLU is defined as $g(x) = \max(0, x)$. For the output layer, we used a linear activation function, defined simply as $g(x) = x$.

Most ANN models of turbulence use a multilayer perception (MLP) network [19,21,22] or some slight variation of an MLP [18,20]. In this work, we also employ an MLP network to implement our model. The network acts on individual grid cells. The network configuration used in this work had an input layer with N_I inputs, a hidden layer with 64 neurons followed by another hidden layer with 32 neurons, and finally an output layer with N_O outputs.

There is some variation in the literature in selecting the input features for ANN models of hydrodynamical turbulence [18–23,25]. The inputs for ANN model τ_{ANN} were all quantities defined for the SGS tensors in Eqs. (25)–(28), the first and second derivatives of those quantities, and the value of all aforementioned terms in cells adjacent to the cell of interest. All derivatives were computed using fourth-order centered finite differencing. For the mixed filtered quantities τ_{ind} and τ_{enth}, we add the mass density ρ to our collection of variables that we include in the inputs in the same manner described above. The inputs to each ANN are explicitly given in Appendix A.

In our case, the outputs are all unique components of the desired SGS tensor which vary depending on the tensor of interest. Thus, we have $N_O = 3$ for τ_{kin} and τ_{mag}, $N_O = 1$ for τ_{ind}, and $N_O = 2$ for τ_{enth}. This differs from most of the literature, where a different ANN is used to find each individual component of the SGS tensor [19,21–23]. By computing all components of the SGS tensor, we hope to incorporate physical symmetries and constraints into future models of τ_{ANN} such as Galilean invariance, though we do not attempt to do so in this work.

For reference, we have chosen mean-squared error as the loss function to optimize the performance of our neural network model. We describe in detail the high-resolution simulations of the magnetized KHI used to train and test our models in Sec. IV. The hyperparameters of our neural network model are presented in Sec. V.

IV. SIMULATION

To train and evaluate the model, we ran 2D magnetized KHI simulations. As described above, the KHI instability occurs when two fluids are moving in opposite directions. When magnetic fields are included, the instability accelerates and the magnetic fields are amplified throughout the process. The KHI was selected because BNS mergers, the targeted application of this work, experience a KHI-like process during the merger phase.

The simulations were run using the open-source SIMFLOWNY code [33,34]. For these simulations, the grid was a Cartesian square with $x, y \in [-L/2, L/2]$, with length $L = 1$. These simulations were performed at three grid sizes with the number of points $N = 512^2$, $N = 1024^2$, and $N = 2048^2$ for the low, medium, and high resolutions, respectively. The boundary conditions were chosen to be periodic in all directions. We evolved the equations for ten units of time. Using a RK4 time integration scheme, we evolved the MHD equations in Eqs. (5)–(8) with time steps of $\Delta t = \frac{0.25}{\sqrt{N}}$. We show density plots of these simulations in Figs. 2 and 3.

To assist in triggering the instability, we add velocity perturbations to the system in both coordinate directions. The specific setup for the initial conditions for the grid functions in this simulation is given by

$$\rho = \rho_0 + \text{sgn}(y)\left[\delta\rho \tanh\left(\frac{|y| - y_l}{a_l}\right)\right], \tag{39}$$

$$v_x = \text{sgn}(y)\left[v_{x0} \tanh\left(\frac{|y| - y_l}{a_l}\right)\right] + \delta v_x \sin(2\pi n_x y), \tag{40}$$

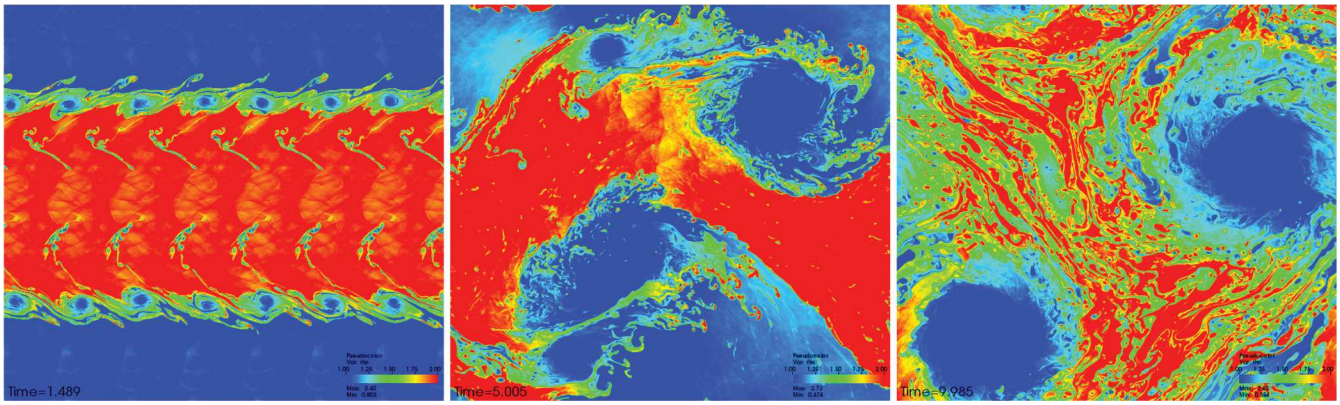

FIG. 2. Plots of the mass density distribution ρ time slices at $t = 1.5$, 5, 10 of our magnetized KHI simulations with a resolution of $N = 2048^2$. On the left, we have the $t = 1.5$ time slice, in which we can observe vortexes begin to form between the two fluids. The number and size of these vortexes are controlled by an initial sinusoidal perturbation of the fluid velocity in the y direction. The center plot at $t = 5$ occurs after many of the aforementioned vortexes have merged together, at which point the flow has become unsteady. This unsteady flow manifests itself in the high-density fluid beginning to break apart. The rightmost plot at $t = 10$ depicts the fluids in a turbulent mixing process with two low-density vortexes helping to drive this mixing.

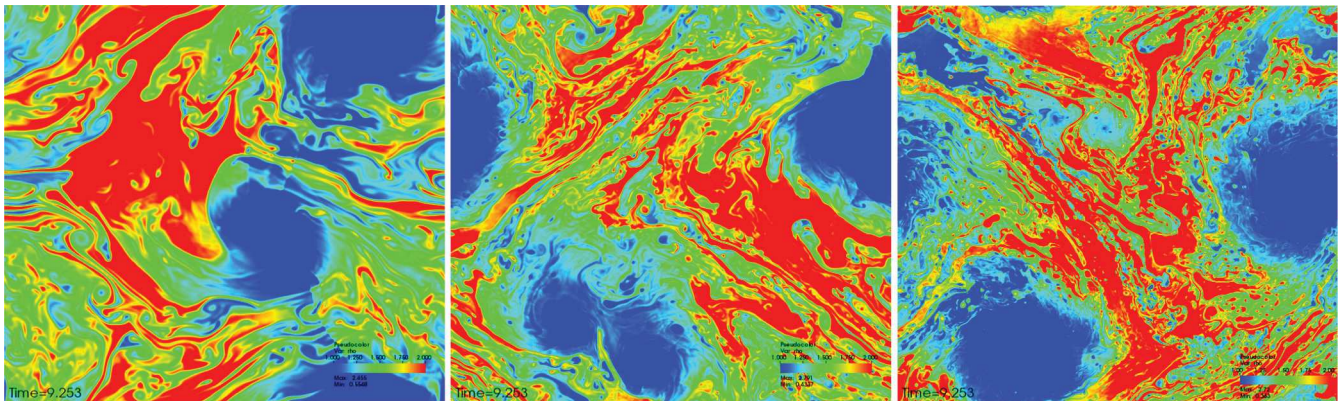

FIG. 3. Plots of the density distribution ρ of the test dataset at time slices $t = 9.25$ for resolutions $N = 512^2$, 1048^2, and 2048^2 in the left, middle, and right image, respectively. All three datasets appear to be undergoing a turbulent mixing process at $t = 9.25$. We observe that, while the $N = 1024^2$ and $N = 2048^2$ runs appear to share many of the same general characteristics, the $N = 512^2$ run fails to reproduce these same features. This failure implies that $N = 512^2$ is not enough to capture the turbulent effects of the magnetized KHI without a SGS model. We note that the testing time slice is of particular importance, because it was used to evaluate the SGS models.

$$v_y = \mathrm{sgn}(y)\left\{\delta v_y \sin(2\pi n_y x)\exp\left[-\left(\frac{|y| - y_l}{\sigma}\right)^2\right]\right\}, \quad (41)$$

$$B_x = B_{x0}, \quad (42)$$

$$B_y = B_{y0}, \quad (43)$$

$$p = p_0. \quad (44)$$

In the above expressions, $\rho_0 = 1.5$ and $\delta\rho = -0.5$ are the average and difference of the low-density region $\rho_1 = 1$ and high-density region $\rho_2 = 2$, respectively. $y_l = 0.25$ is the y coordinate where the transition from ρ_1 to ρ_2 occurs. $a_l = 0.01$ is the characteristic size of this transition region, providing a smooth transition that mitigates some of the numerical instabilities of the transition between the different density regions. $v_{x0} = 0.5$ is the initial velocity of the fluid in the x direction. $\delta v_x = 0.01$ is a sinusoidal perturbation of v_{x0} with $n_x = 4$ periods going along the y direction. $\delta v_y = 0.2$ is a sinusoidal perturbation of the y component of the velocity with $n_y = 7$ periods along the x direction. $\sigma = 0.1$ is the characteristic Gaussian falloff of δv_y away from y_l. We note that, for $|y| > 0.45$, δv_y is set to 0. $B_{x0} = 0.001$, $B_{y0} = 0$, and $p_0 = 1$ are the initial x component of the magnetic field, initial y component of the magnetic field, and initial pressure, respectively.

Like Ref. [13], we desired to evolve with similar numerical methods to those used in numerical relativity simulations of BNS mergers. We employed the method of lines to discretize our system of equations. We used a finite volume scheme with MP5 reconstruction and local lax Friedrichs flux splitting for the evolution of our system, which provides numerical stability even in the presence of shocks. This scheme views the ideal MHD equations in Eqs. (5)–(8) as

$$\partial_t \boldsymbol{U} + \partial_i \boldsymbol{F}^i = \boldsymbol{S}, \quad (45)$$

where $\boldsymbol{U} = \{\rho, \rho v^j, B^j, u\}$ are our conserved quantities, $\boldsymbol{F}$ are the fluxes for those conserved fields, and $\boldsymbol{S} = \{0, 0, 0, 0\}$ are the source terms. The source term is set to zero in our case but is nonzero, in general, if, say, an external force like gravity is applied to the fluid. $\boldsymbol{F}$ is allowed to depend on the conserved variables but not on their derivatives. The SGS tensors, which depend on derivatives of the conserved variables, would be placed in $\boldsymbol{S}$ rather than $\boldsymbol{F}$ when implementing one of the aforementioned SGS models in a simulation.

To preserve the divergence-free condition on the magnetic field, we used a hyperbolic divergence cleaning [35]. This divergence cleaning adds another evolution equation to our system for ϕ to ensure the magnetic field divergence decays to 0 and is defined as

$$\partial_t \phi + c_h^2 \partial_i B^i = -\frac{c_h}{c_r}\phi, \quad (46)$$

where $c_h = 1$ and $c_r = 0.18$.

V. TRAINING

The KHI simulation data were filtered using a box filter with filter sizes $f = 2$, 4, 8, 16, where f is defined as $f = \frac{\Delta_f}{\Delta}$. For each of the filter sizes and resolutions, SGS tensors and inputs to the ANNs were calculated after $t = 1$ every ∼0.1 time units until the simulation ended at $t = 10$.[1] The test data evaluated these same quantities at $t \approx 9.25$.

[1]Because of memory-consumption issues, we used less data to train the $N = 1024^2$ $f = 2$ and $N = 2048^2$ $f = 2, 4$ models. Specifically, the $N = 1024^2$ $f = 2$ and $N = 2048^2$ $f = 4$ models sampled training data every ∼0.5 time units. The $N = 2048^2$ $f = 2$ model sampled training data every ∼1 time units.

This approach ensures that even models that are trained with low-resolution simulations are exposed to data with sufficient size and variety. We found that this approach prevents overfitting. In 3D, we expect to use fewer time slices, as each time slice contains significantly more samples than in 2D. Another observation is that we experimented with data augmentation methods, as those described in Ref. [19], which consist of augmenting the data by providing multiple copies of each time slice but choosing a different point after filtering [19]. However, we found that this approach does not generalize well during testing. To address that problem, we chose multiple time slices during training.

After calculating the SGS tensors and the necessary inputs to the ANN models, we exported the data to train the model in TensorFlow [36]. The data were normalized to have zero mean and unit standard deviation. We used 10% of the simulated data for validation purposes. For the training of the neural network model, we used an ADAM optimizer with early stopping [37]. The maximum number of epochs was 100. A batch size of 1000 was used during training.

VI. METHODOLOGY

In this section, we describe quantities that we will use to test our neural network model and metrics to assess its ability to correctly reproduce true features and properties of the testing dataset.

A. Spectra calculation

The energy spectrum $\mathcal{E}(k)$ represents the spatial scale at which the energy is distributed in a given process. For low wave number k, we see the large-scale features of the energy spectrum. On the other hand, high k values give the small-scale features of the spectrum. The ultimate goal of the large eddy simulation is to reproduce the energy spectrum of the DNS simulations as closely as possible. Appendix B describes how to compute these quantities.

In MHD turbulence, we are concerned about the energy spectra of the kinematic motion $\mathcal{E}_{\mathrm{kin}}(k)$ and the magnetic field $\mathcal{E}_{\mathrm{mag}}(k)$. We note that these energy spectra have a different expected distribution. The kinetic energy spectrum falls off as $\mathcal{E}_{\mathrm{kin}}(k) \propto k^{-5/3}$ at high wave numbers. However, the magnetic energy spectrum rises as $\mathcal{E}_{\mathrm{mag}}(k) \propto k^{3/2}$ at large k values [13]. Thus, we expect the small-scale behavior will be especially significant in the overall magnetic energy contribution and must be modeled carefully.

Moreover, we are interested in the total energy obtained by integrating over all the spectra. By examining how the total energy changes over time, we can extract useful information about characteristics of the simulation. In particular, we would like to measure how the kinetic energy E_{kin} and magnetic energy E_{mag} change through the effect of the KHI.

B. Model performance criteria

To quantify the performance of our models, we will use several common turbulence statistics. The first of these statistics is the correlation coefficient C, which shows how well the data and the model follow one another. We define C as

$$C = \frac{\langle (\tau_{\mathrm{DNS}} - \langle \tau_{\mathrm{DNS}} \rangle)(\tau_{\mathrm{model}} - \langle \tau_{\mathrm{model}} \rangle) \rangle}{\sqrt{\langle (\tau_{\mathrm{DNS}} - \langle \tau_{\mathrm{DNS}} \rangle)^2 \rangle \langle (\tau_{\mathrm{model}} - \langle \tau_{\mathrm{model}} \rangle)^2 \rangle}}, \quad (47)$$

where τ_{DNS} is the SGS tensor computed from filtering the high-resolution data, τ_{model} is the SGS tensor computed from the SGS model we are testing, and $\langle x \rangle$ is the volumetric average of the quantity x. C can range from -1 to 1 with values near to -1 being anticorrelated, values near to 0 being uncorrelated, and values near to 1 being well correlated. Simply put, the closer C is to 1, the better the model. We use C as our primary measure of performance for our models. We will also look at the relative error between the model and the simulation denoted by E. E is defined as

$$E = \frac{\sqrt{\langle (\tau_{\mathrm{DNS}} - \tau_{\mathrm{model}})^2 \rangle}}{\sqrt{\langle \tau_{\mathrm{DNS}}^2 \rangle}}, \quad (48)$$

with all quantities defined in the same manner as in Eq. (47). We note that the lower the value of E is for a model, the better the model. The root mean square (rms) of the model tells us the degree to which the model deviates from the average. The rms of a quantity x is given by

$$\mathrm{rms}(x) = \sqrt{\langle (x - \langle x \rangle)^2 \rangle}. \quad (49)$$

Here, we will calculate the rms for τ_{model} and τ_{DNS}. The goal here is for the rms of τ_{model} is to be as close to the rms of τ_{DNS} as possible. In addition, we would like to use the absolute value of the rms of τ_{DNS} to tell us more about the features of τ for the various models, resolutions, and filter sizes.

VII. RESULTS

In this section, we present results of several tests we conducted to assess the reliability of our neural network model to accurately capture the physics of our testing datasets.

A. Spectra

We will begin the discussion of the results by analyzing the spectra of the simulations. The first three images in Fig. 4 illustrate the spectra at the time slices of the simulation that were featured in the density plots in Fig. 2 with all simulation resolutions included. These selected time slices occur at approximately $t = 1.5$,

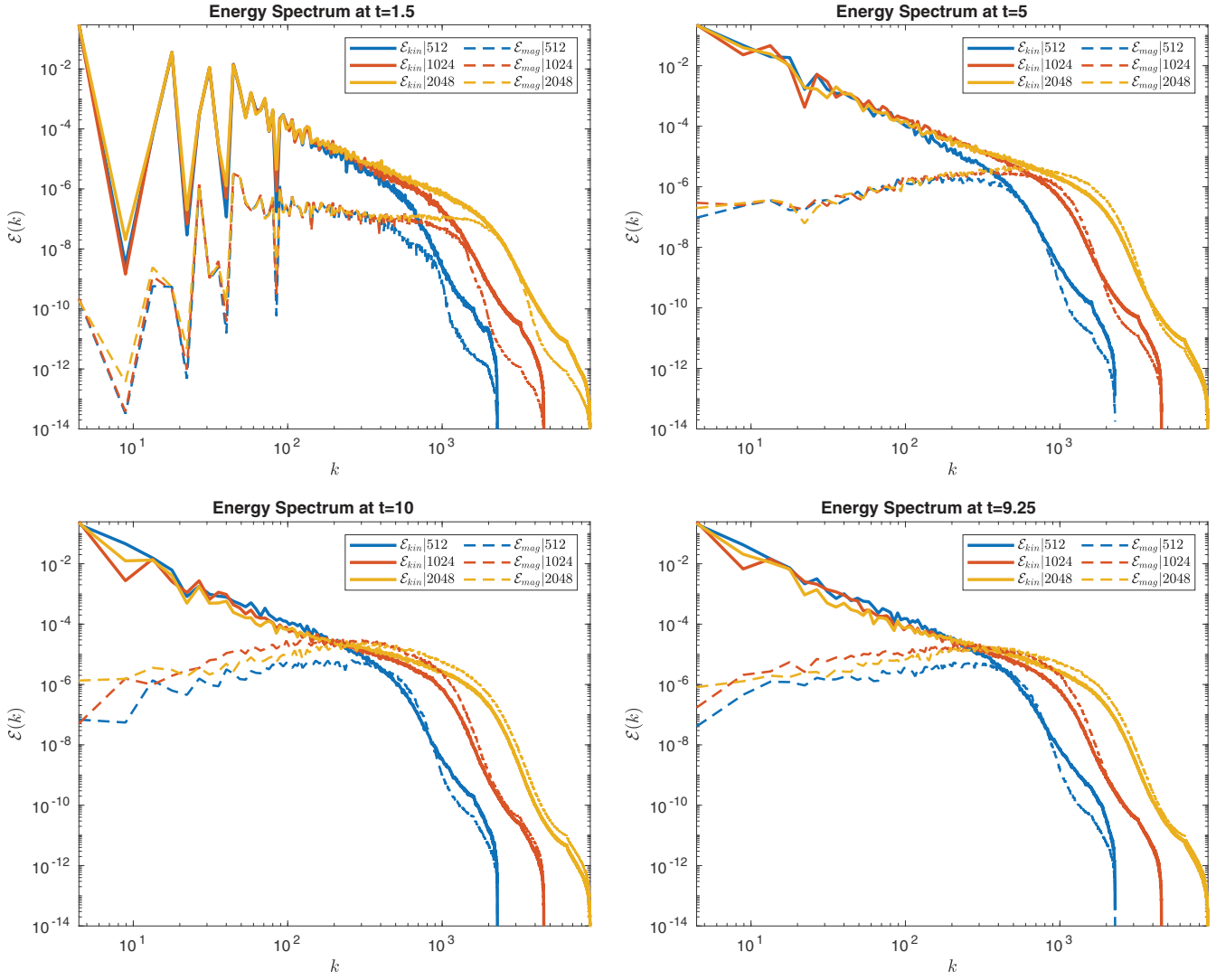

FIG. 4. Plots of the energy spectra $\mathcal{E}(k)$ at various time steps throughout the simulation for each resolution. The kinetic energy spectra $\mathcal{E}_{\text{kin}}(k)$ is given by the solid lines, while the magnetic energy spectra $\mathcal{E}_{\text{mag}}(k)$ is denoted by the dashed lines. The resolutions $N = 512^2$, 1024^2, and 2048^2 are given by the blue, red, and green lines, respectively. The top left, top right, and bottom left images are taken at approximately $t = 1.5$, 5, and 10, respectively, and correspond to the time steps displayed of the density distribution plots in Fig. 2. The bottom right plot provides the spectra of the test dataset used for evaluating the models whose density distribution is featured in Fig. 3.

5, 10. The last plot in Fig. 4 depicts the spectrum of the test dataset whose density distribution can be seen in Fig. 3.

Figure 5 includes both $\mathcal{E}_{\text{kin}}$ and $\mathcal{E}_{\text{mag}}$. We observe that the $\mathcal{E}_{\text{kin}}$ of the plots is fairly similar at low k values. The obvious exception to this is the $t = 1.5$ plot, where the low k spectrum appears to still be settling down for both energy types, though this effect does not appear to be resolution dependent. We also notice that the 2048^2 resolution simulation has reduced values of $\mathcal{E}_{\text{kin}}$ at low k compared to the other simulations at later times, likely due to the kinetic energy being converted into magnetic energy more efficiently at high resolutions. At high k values, we observe a faster $\mathcal{E}_{\text{kin}}$ falloff at low resolution. This drop off is likely due to the effect of the finite grid resolution on the small-scale features.

The magnetic field spectrum at low k is significantly smaller than its kinetic energy counterpart. As k increases, the magnetic field spectrum increases; it may eventually surpass the kinetic energy spectra before decaying. It appears that much of this decay is an effect of the finite grid resolution. At later times, all $\mathcal{E}_{\text{mag}}$ spectra increase considerably. The high-resolution simulations have noticeably greater $\mathcal{E}_{\text{mag}}$ than those at lower resolutions. This effect is likely caused by the conversion of kinetic energy to magnetic energy being more efficient at high resolutions.

The last plot in Fig. 4 presents the spectra of our testing dataset. We observe that this plot shares similar characteristics with the $t = 10$ spectra plot. However, we note that the high k region of the $N = 1024^2$ simulation's $\mathcal{E}_{\text{mag}}$

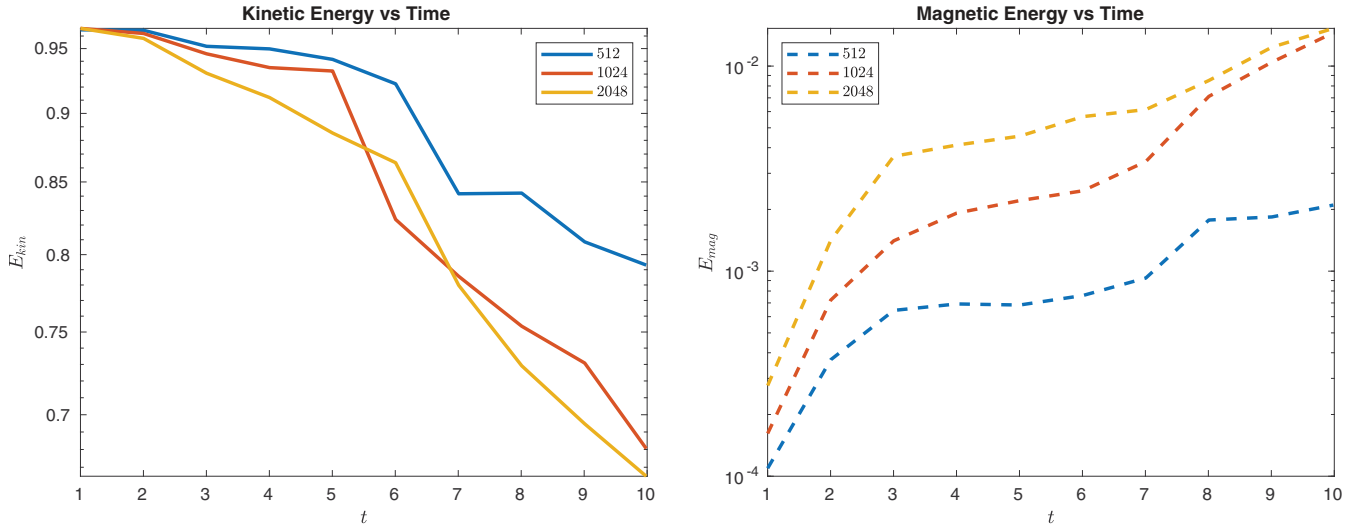

FIG. 5. Time evolution of the total kinetic energy E_{kin} (left panel) and total magnetic energy E_{mag} (right panel). The kinetic energy decreases over time as it is being converted into magnetic energy. This energy conversion can be seen in the right panel, where E_{mag} increases over time. We notice that E_{kin} decreases more rapidly for the high-resolution runs while E_{mag} increases more rapidly. This would indicate that this conversion from E_{kin} to E_{mag} occurs most efficiently at small scales that high-resolution simulations can best resolve.

spectra is weaker relative to the $N = 2048$ simulation's $\mathcal{E}_{\mathrm{mag}}$ spectra than in the $t = 10$ spectra plot.

Figure 5 shows the integrated energy spectrum or total energy vs time starting at $t = 1$ for both the kinetic energy E_{kin} and the magnetic energy E_{mag} in the first and second plots, respectively. We notice that E_{kin} starts the same for all simulations but decreases over time. The higher-resolution simulations decreased in E_{kin} faster than those at lower resolutions. This may indicate that E_{kin} is being converted into E_{mag}. On the other hand, E_{mag} started fairly similar in magnitude for all resolutions with deviations of the order of unity. We then see an increase in the magnetic energy with the higher-resolution simulations increasing much faster than their lower-resolution counterparts. At $t \sim 7$, the $N = 1024^2$ simulation is observed to rise faster than the $N = 2048^2$ simulation, which results in both simulations having nearly equal energy by the end of the simulation at $t = 10$. E_{mag} still appears to be increasing at $t = 10$, which may indicate that the process of magnetic amplification may still be ongoing.

B. Model performance

For all subgrid filter sizes f and at all resolutions N, our findings indicate that our neural network model outperforms the gradient model when evaluated on the test data. To show this, let us first take a look at the results of models with $N = 2048^2$ at $f = 8$, a case where the differences can be clearly observed between the two SGS models.

Figure 6 presents targets vs predictions of the SGS models for the test data at $N = 2048^2$ and $f = 8$. We notice that both models show good performance when the magnitude of the SGS tensor is low. However, at high SGS tensor magnitudes, the gradient model significantly underestimates the SGS quantities. Compared to the gradient model, the ANN models predict more accurate values for those high-magnitude targets, in particular, for the components of τ_{mag} and τ_{ind} tensors. We note that one reason for the poor performance of the gradient model for high SGS tensors may be due to it being a first-order model and could potentially be improved using higher-order corrections.

In Fig. 7, we show a histogram of the normalized distribution of the SGS tensors of the $N = 2048^2$, $f = 8$ test data as well as those predicted by the ANN and gradient models for this same dataset. We find that, overall, the ANN model's predictions more closely resemble the distribution calculated from the DNS dataset compared to those of the gradient model. This improvement is especially noticeable for the τ_{mag} tensor components. We also notice that the ANN model predicts some negative values for the diagonal components of the SGS tensors τ_{kin} and τ_{mag}. However, these predictions are unphysical, as they violate the realizability constraint which requires $\tau_{ii} \geq 0$ [38,39]. Such unphysical behavior can be resolved in ANN models by embedding the physical constraint in the loss function, which will be a subject of future work.

To understand exactly how models behave for a single time slice of data, Fig. 8 shows the values of the components of the SGS tensor τ_{mag} for the actual DNS data τ_{DNS}, the ANN model τ_{ANN}, and the gradient model τ_{grad}. From these plots, we observe that τ_{ANN} performs noticeably better in regions with significant small-scale

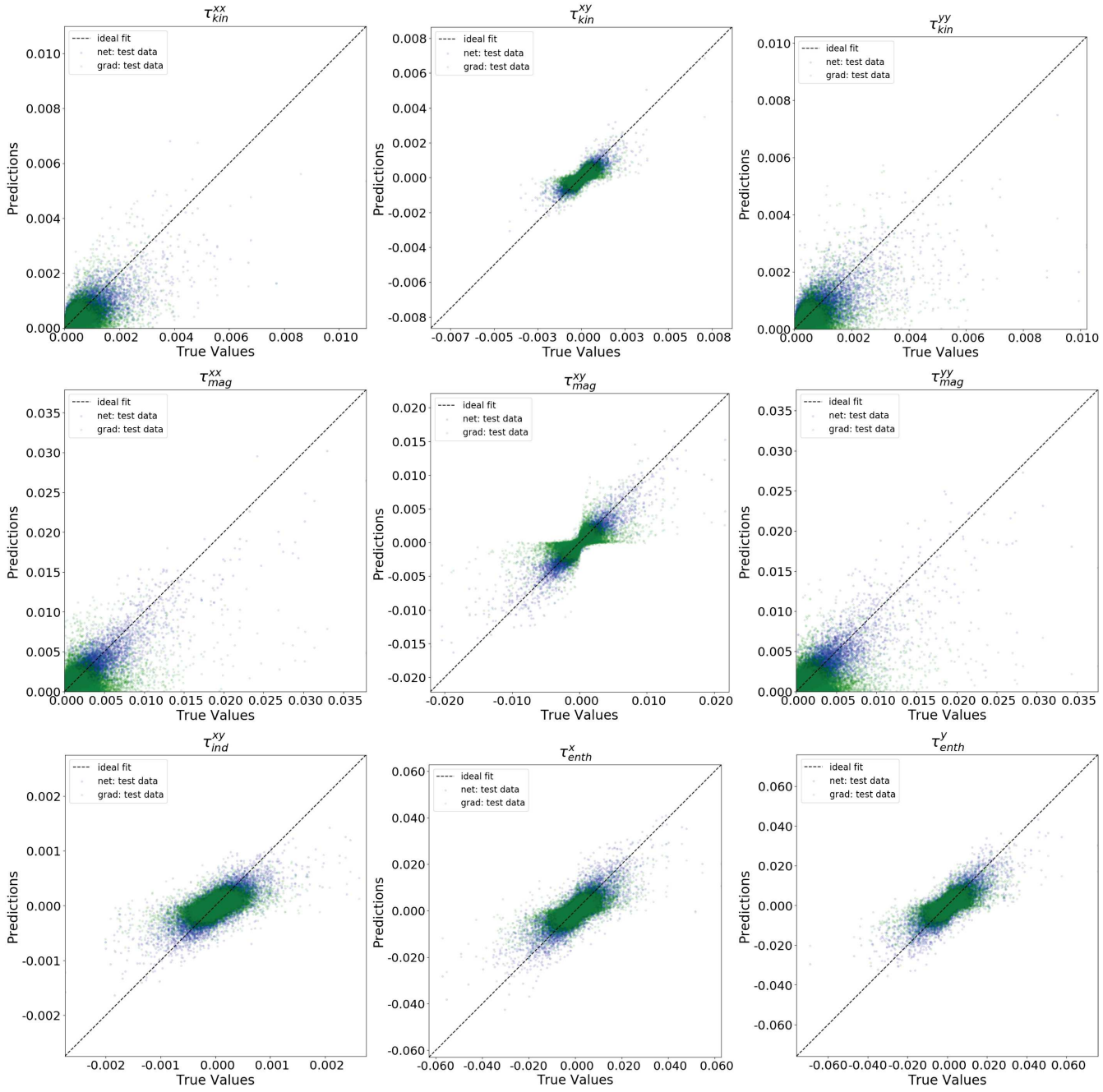

FIG. 6. Predictions vs target values of SGS tensors for the $N = 2048^2$ resolution test dataset with filter size $f = 8$ for each of our SGS models. The small transparent green circles represent values of the gradient model and are overlaid on top of the small transparent blue circles, which represent the values of the ANN model. The black dashed line depicts a perfect one-to-one matching between the targets and the predictions. The first row displays the xx, xy, and yy components of τ_{kin} from left to right, respectively. The middle row shows the same information for the components of the τ_{mag} SGS tensor. The bottom row shows the xy component of τ_{ind} on the left, the x component of τ_{enth} in the middle, and the y component of τ_{enth} on the right. We observe that the blue circles of the ANN model appear much closer to the dashed black line for high SGS tensor values than the green circles of the gradient model, especially in the τ_{mag} and τ_{ind} SGS tensors.

structure in τ_{DNS} compared to τ_{grad}. This effect is most prevalent in τ_{mag} in Fig. 8, though is visible for most of the other tensors. The plots depicting the values of the other SGS tensors can be seen in Figs. 12–14 in Appendix C.

Having examined a specific SGS tensor qualitatively, we will now move toward a more general quantitative discussion of the behavior of the SGS tensors at different resolution and filter sizes for the models. For this, we will

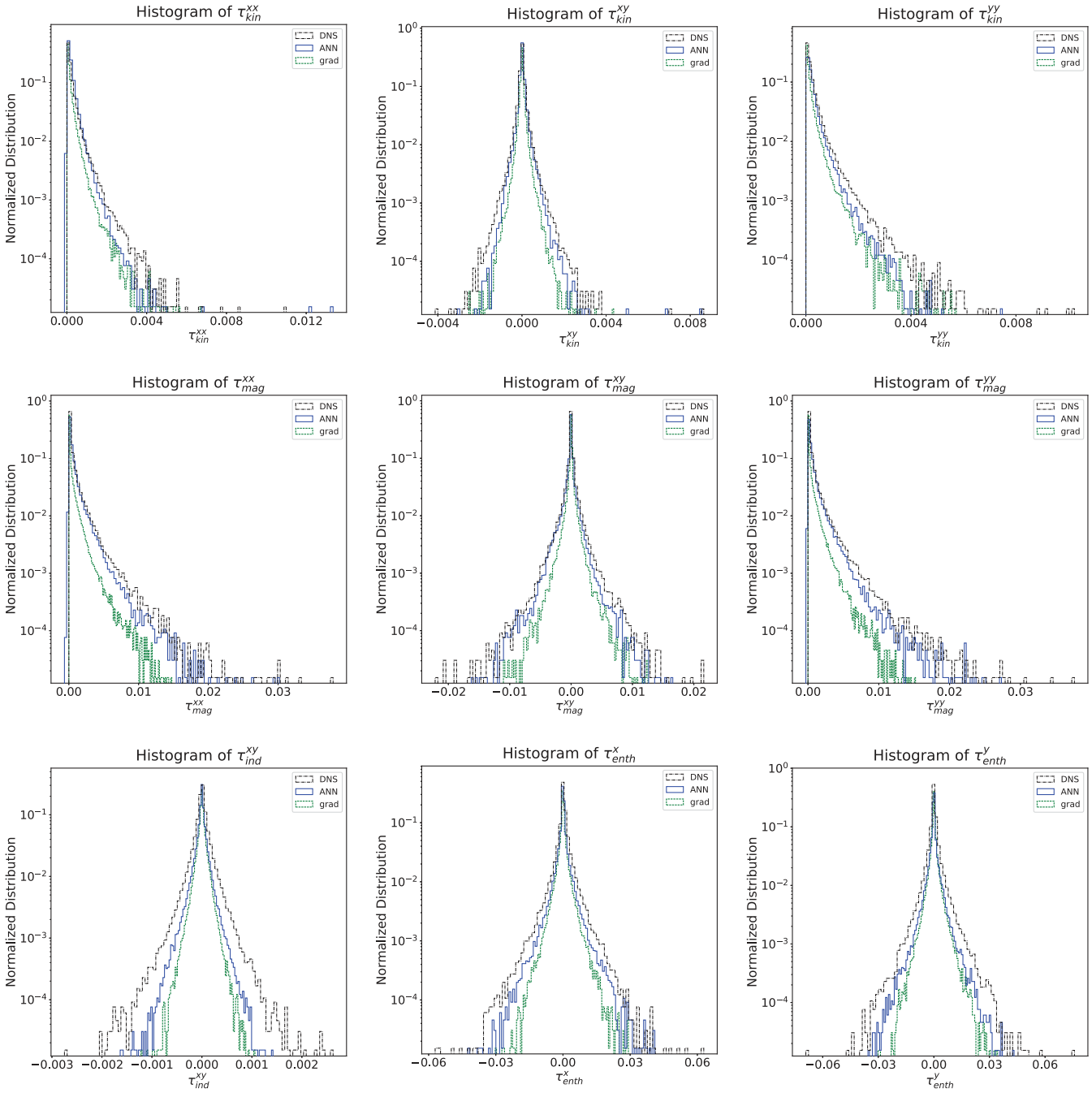

FIG. 7. Histogram of the normalized probability distribution of the values of SGS tensors for the $N = 2048^2$ resolution test dataset with filter size $f = 8$ for each of our SGS models. The black dot-dashed line represents the distribution of the DNS data, the solid blue line represents the distribution of the ANN model predictions, and the green dashed line represents the distribution of the gradient model predictions. The first row displays the xx, xy, and yy components of τ_{kin} from left to right, respectively. The middle row shows the same information for the components of the τ_{mag} SGS tensor. The bottom row shows the xy component of τ_{ind} on the left, the x component of τ_{enth} in the middle, and the y component of τ_{enth} on the right. We observe that the ANN model predictions more closely resemble the distribution of the DNS values than those of the gradient model.

start by looking at the correlation coefficient C presented in Fig. 9.

In Fig. 9, we show the plots of correlation coefficient C vs the filter size f for all resolutions simulated in this study. Our findings show that all ANN models performed better than their gradient model counterparts for every SGS tensor component at the same N and f. The degree to which this improvement occurred was dependent primarily on the

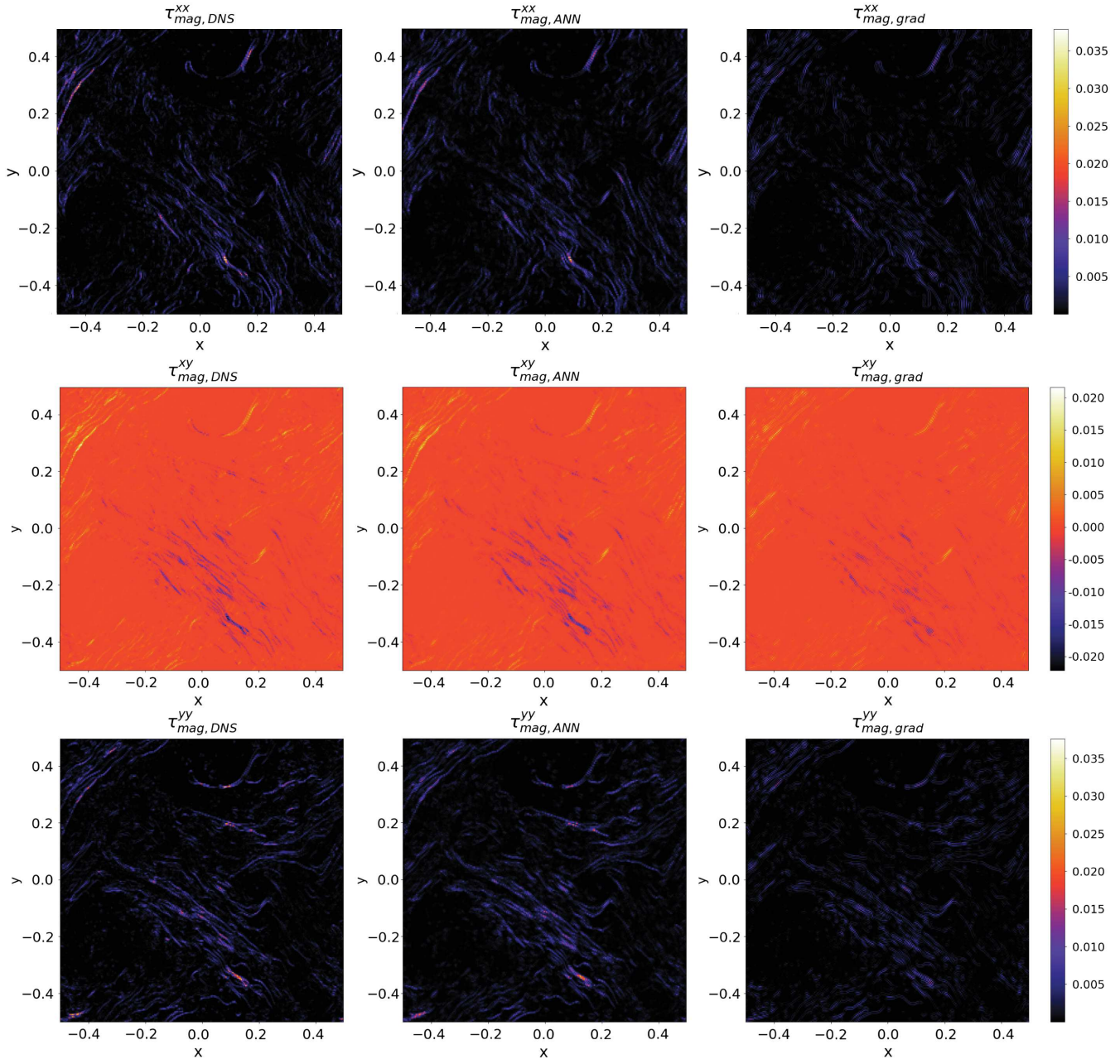

FIG. 8. Plots of the components of the τ_{mag} SGS tensor of the test dataset for the $N = 2048^2$ resolution run with a filter size of $f = 8$. The columns depict the SGS tensor values of the exact DNS calculation, the ANN model reconstruction, and the gradient model reconstruction from left to right, respectively. The top, middle, and bottom rows display the xx, xy, and yy components, respectively. We observe that, while the ANN model appears to reproduce most of the visual features of the DNS calculation, the gradient model appears to struggle in regions with more detailed structure.

filter size and the SGS tensor being analyzed. The effect of the resolution is not entirely clear, but both models appear to follow similar trajectories on lines at the same resolution. In general, the value of C decreased as f increased. This was particularly prevalent in the τ_{mag} and τ_{ind} tensors. This decrease in C at high f was much more significant in τ_{grad} than in τ_{ANN}. This indicates that τ_{ANN} performs better at higher filter sizes, implying that we would be able get accurate results from employing the ANN models at lower resolutions than we could from the gradient model.

Moreover, the gradient model's difficulty calculating τ_{mag} and τ_{ind} at high filter sizes suggests that it is not able to reproduce the effects of turbulence on the magnetic fields at lower grid resolutions. In contrast, our results indicate that neural networks can address these limitations in an *a posteriori* study. We observed that the purely

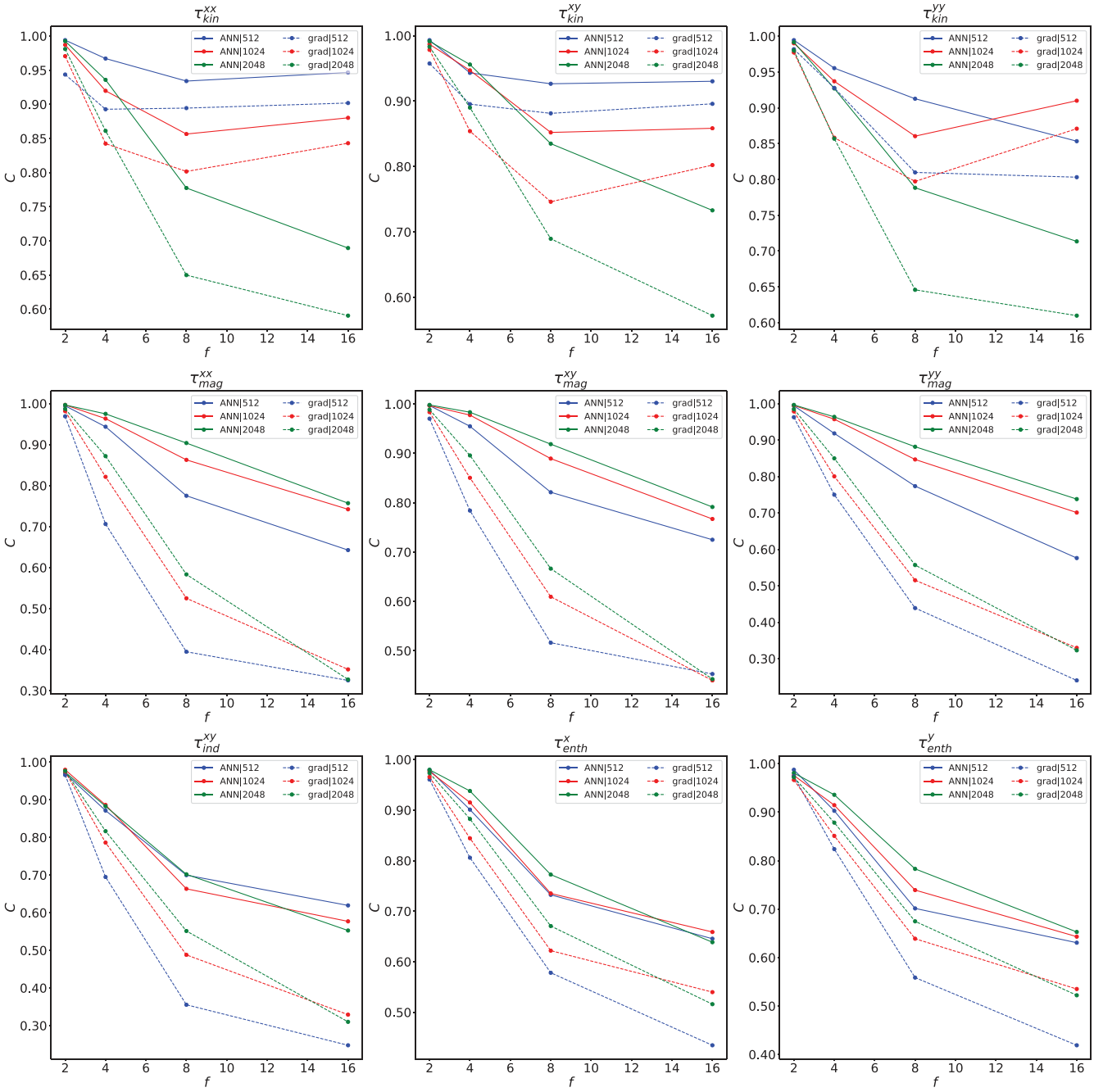

FIG. 9. Here we plot the correlation coefficient C as a function of filter size f for all resolutions N, SGS tensor components τ, and SGS models. The solid lines refer to the ANN model, and the dashed lines refer to the gradient model. The resolutions are given by the color of the line; blue represents the $N = 512^2$ simulation, red represents the $N = 1024^2$ simulation, and green represents the $N = 2048^2$ simulation. We observe that the ANN model has a higher correlation coefficient than the gradient model for all SGS tensor components at all resolutions. We also note that C generally decreases with increasing f, but this decay affects the gradient model more significantly.

hydrodynamical τ_{kin} SGS tensor was the easiest to compute accurately for both the ANN and gradient models. Thus, the improvements in the ANN model's calculation of τ_{kin} should be considered less beneficial than those from τ_{mag} and τ_{ind}.

The energy SGS tensor τ_{enth} also receives a noticeable improvement from the use of the ANN model over the gradient model. This effect is again most prevalent at high f values, more than for τ_{kin} but not quite as significant as the τ_{mag} or τ_{ind} terms. We should note again that the gradient model is a leading-order expansion of the filtering operator in grid spacing, which corresponds to filter size f. With a higher-order expansion of the filtering operator, we may see some improvement at high f.

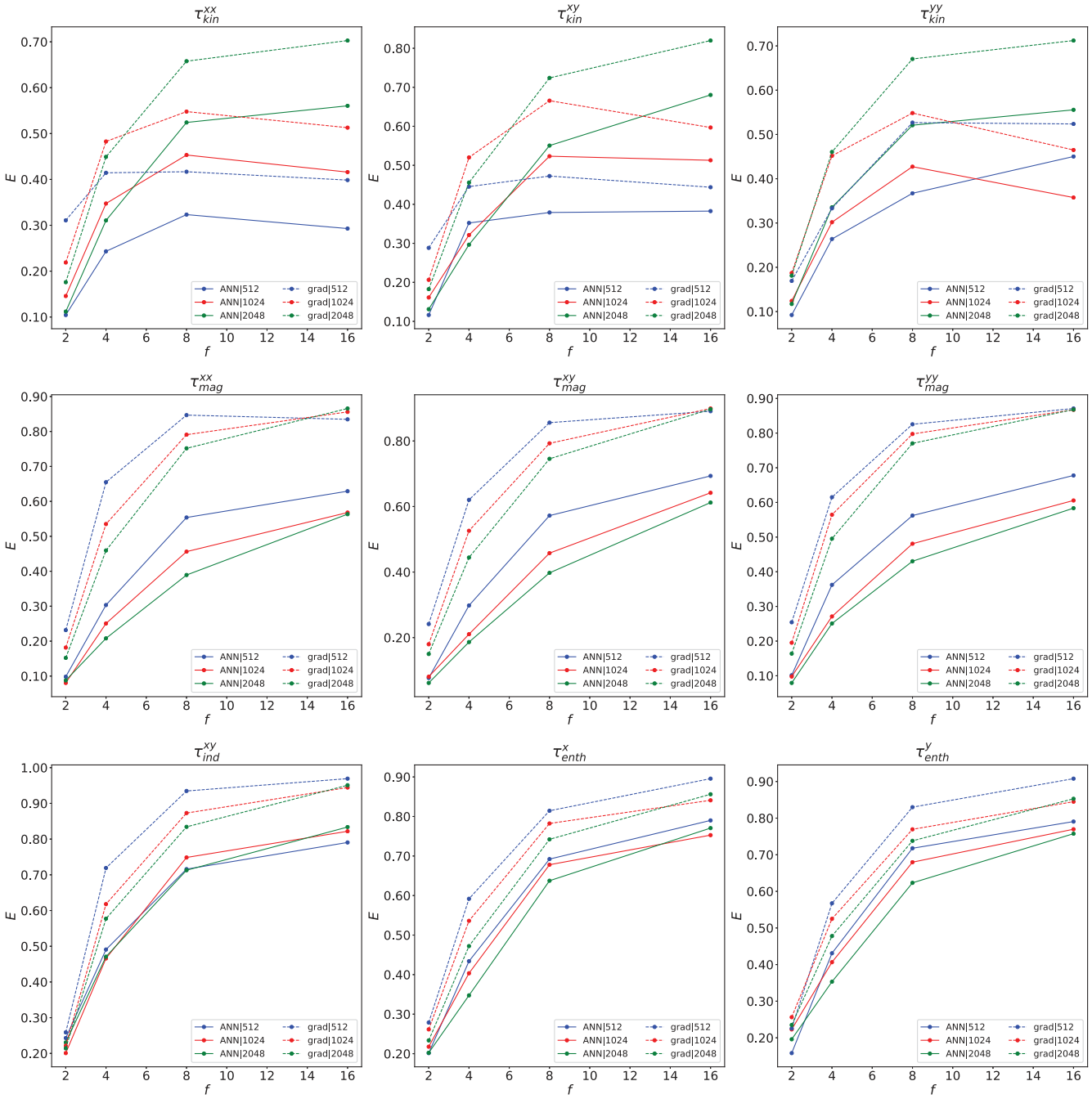

FIG. 10. Here we plot the relative error E as a function of filter size f for all resolutions N, SGS tensor components τ, and SGS models. The solid lines refer to the ANN model, and the dashed lines refer to the gradient model. The resolutions are given by the color of the line; blue represents the $N = 512^2$ simulation, red represents the $N = 1024^2$ simulation, and green represents the $N = 2048^2$ simulation. We observe that E is lower for the ANN model than for the gradient model for all SGS tensor components at all resolutions. We also note that E generally increases with f but is more severe for the gradient model than for the ANN model.

Figure 10 shows the relative error E between the predictions of the gradient and ANN models compared to the DNS data. The results mirror those discussed for the correlation coefficient C in terms of E increasing with f more quickly for the gradient model compared to the ANN model.

In Fig. 11, we show the rms of τ_{DNS}, τ_{ANN}, and τ_{grad} for all SGS tensor components at all resolutions. For τ_{kin} we observe that the two models perform similarly in terms of their proximity to the rms of τ_{DNS}, and both slightly undershoot the true value for this SGS tensor. We also notice that the value of the rms increases with filter size f,

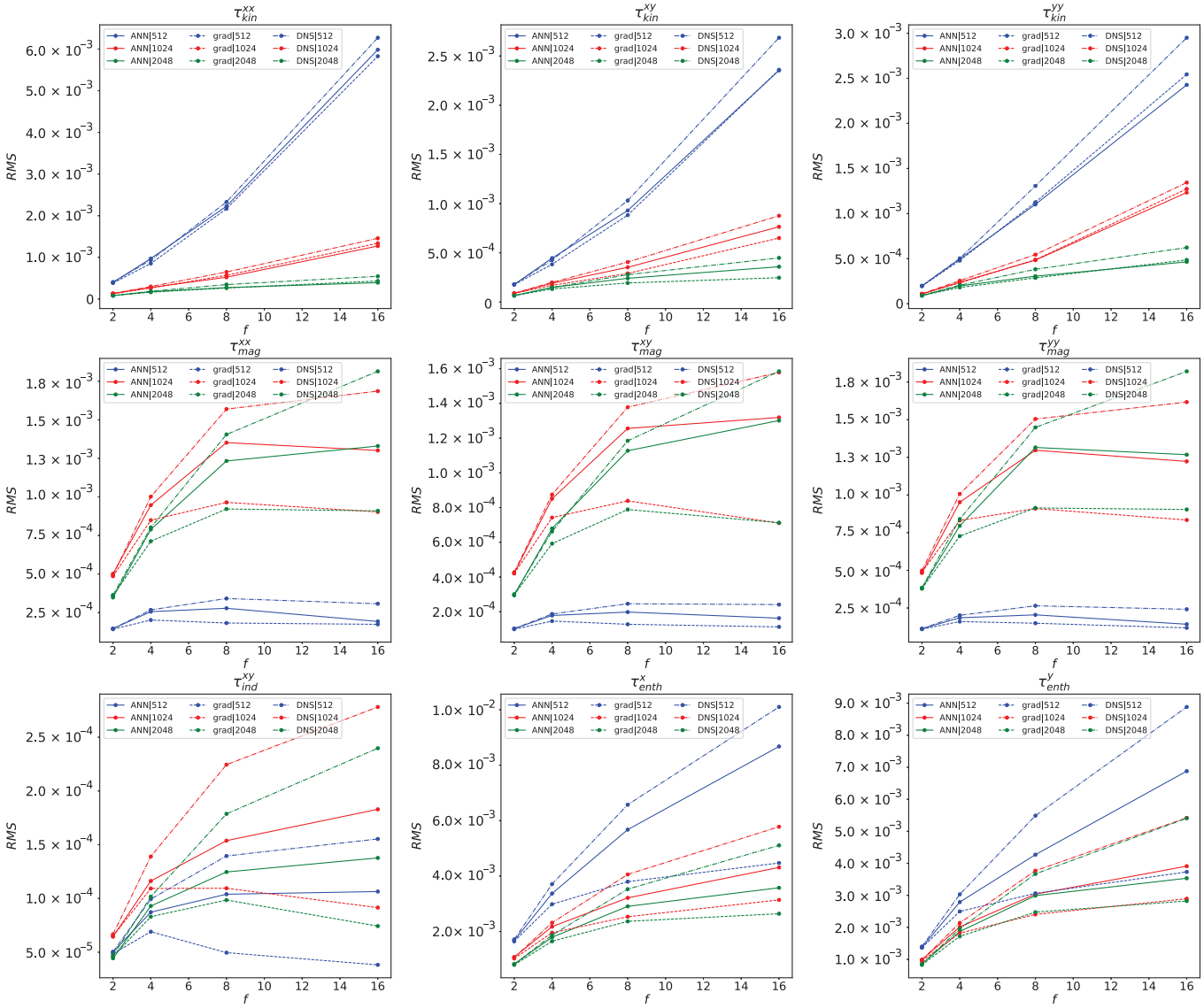

FIG. 11. Here we plot the rms value as a function of filter size f for all resolutions N, SGS tensor components τ, and SGS models including the DNS calculation. The solid lines refer to the ANN model, the dashed lines refer to the gradient model, and the dash-dotted lines refer to the DNS calculation. The resolutions are given by the color of the line; blue represents the $N = 512^2$ simulation, red represents the $N = 1024^2$ simulation, and green represents the $N = 2048^2$ simulation. We observe that the rms values are fairly similar for the τ_{kin} SGS tensor components (top row) for both SGS models and the DNS result. For the SGS tensors τ_{mag} (middle row), τ_{ind} (bottom right), and τ_{enth} (bottom middle and left), the ANN model has a much closer rms value to the DNS result compared to the gradient model.

while the performance of both models decreases slightly for this metric at high f for τ_{kin}. This makes sense, as one would expect more SGS behavior at high f as the grid increases we have more SGS phenomena, resulting in the SGS tensors being more difficult to model. The rms of the low-resolution data is greater than that of the high-resolution simulations for τ_{kin}. Moreover, this rise in rms appears to be polynomial in f and occurs much more prevalently at lower resolutions. This would imply that there does not exist a preferred scale for τ_{kin}, as the rms value appears to increase with the volume of the grid.

The rms plots of τ_{mag}, on the other hand, demonstrate clear differences in the behavior of the models. For example, the rms of the ANN model is considerably higher than that of the gradient model for all components of τ_{mag}. In turn, the rms of τ_{ANN} is much closer than τ_{grad} to the rms of τ_{DNS}, which is greater than either model. The difference between the rms of τ_{model} and the rms of τ_{DNS} is greatest at

high values of f for both the ANN and gradient models, with the difference between the rms of $\tau_{\rm DNS}$ and $\tau_{\rm grad}$ being much more severe. We also observe the $N = 512^2$ resolution runs have a significantly lower rms value than the others, implying that this resolution was too low to capture much of the SGS behavior of the magnetic field. Meanwhile, the rms values of the $N = 1024^2$ and the $N = 2048^2$ simulations are fairly similar, which is not surprising given that the value of $E_{\rm mag}$ is fairly similar for the two resolutions at $t = 9.25$, the time slice of the training data. The rms value increases with f for $\tau_{\rm mag}$, but this increase slows down at high f. In contrast, we recall that the increase in rms of $\tau_{\rm kin}$ accelerates at high values of f. This behavior when taken in conjunction with the significantly lower rms value of the lowest-resolution run implies that $\tau_{\rm mag}$ prefers to act on small scales.

The plots of the rms values of $\tau_{\rm ind}$ share many of the same characteristics as those of $\tau_{\rm mag}$ in terms of the superior performance of the ANN model over the gradient model, the rms of the lowest-resolution run having the lowest value, and the deceleration of the increase in rms at a high f value. However, we would like to emphasize that the rms of the $N = 1024^2$ run is clearly greater than the $N = 2048^2$ run unlike $\tau_{\rm mag}$, where their values were fairly similar. This phenomenon likely results from the quick acceleration of the increase of $E_{\rm mag}$ of the test data slice at $t = 9.25$ that was observed in Fig. 5, which is evidenced in $\tau_{\rm ind}$ representing the turbulent amplification of the magnetic field. We also observe that the gradient model performs particularly poorly for this SGS tensor at high f, where the rms of $\tau_{\rm grad}$ actually decreases despite the rms of $\tau_{\rm DNS}$ actually increasing, albeit at a slower rate.

The rms plots of $\tau_{\rm enth}$ show the lowest resolution having the highest rms value, followed by the middle resolution, and then the high resolution as in the plots of $\tau_{\rm kin}$. However, we still observe a slower increase of rms at high f like $\tau_{\rm mag}$ and $\tau_{\rm ind}$. The ANN model clearly models the rms of $\tau_{\rm enth}$ more closely than the gradient model as well. As with all the previously mentioned SGS tensor components, the performance of the models in computing the rms decreases at high f.

VIII. CONCLUSIONS

We performed an *a priori* study to evaluate the accuracy of ANN models of SGS ideal MHD turbulence with high-resolution 2D simulations of the magnetized KHI. This is the first such study of ANNs in MHD turbulence. We compared the performance of the model to the gradient model that has been proposed in similar studies of modeling MHD turbulence in the LES framework.

In this study, we showed that the ANN performs significantly better than the gradient model in reproducing the SGS tensors compared to the gradient model. This improvement occurred at all resolutions, for all SGS tensors, and filter sizes. However, the degree of improvement varied considerably with the SGS tensor and filter size.

In particular, $\tau_{\rm mag}$, representing the turbulent effect of the magnetic field on the motion of the field, and $\tau_{\rm ind}$, representing the turbulent amplification of the magnetic field, are modeled much more accurately than with the gradient model. This allows ANNs to provide a better model of the turbulent effects of the magnetic field than any model in the MHD turbulence literature.

Moreover, we demonstrated that the gradient model's performance falls off significantly at high filter sizes. However, the ANN is able to maintain a much higher correlation coefficient at high filter sizes. This implies that ANNs may be able to reproduce the effect of turbulence more accurately than gradient models.

Having established the potential of these ANN models of MHD turbulence in an *a priori* study, there are various pathways of future study for the use of ANN models. The most obvious of which is *a posteriori* study, where we deploy these models in an actual simulation and quantify how well the SGS models reproduce the spectra. One may also consider evaluating these models for more computationally intensive 3D simulations and, eventually, general relativistic MHD. This work will also require the development of loss functions that incorporate physical constraints such as rotational invariance. These studies will be pursued in the near future.

ACKNOWLEDGMENTS

E. A. H. gratefully acknowledges National Science Foundation (NSF) Grants No. OAC-1931561 and No. OAC-1934757. We are grateful to NVIDIA, for donating several Tesla P100 and V100 GPUs that we used for our analysis, and NSF Grants No. NSF-1550514, No. NSF-1659702, and No. TG-PHY160053. We thank the NCSA Gravity Group for useful feedback.

APPENDIX A: ANN MODEL INPUTS

Here we explicitly note the inputs to each SGS tensor of $\tau_{\rm ANN}$ for clarity:

$$I_{\tau_{\rm kin,net}} = \{\tilde{v}_i^{m,n}, \tilde{v}_i^{m\pm1,n}, \tilde{v}_i^{m,n\pm1}, \partial_p \tilde{v}_i^{m,n}, \partial_p \tilde{v}_i^{m\pm1,n}, \partial_p \tilde{v}_i^{m,n\pm1}, \partial_p\partial_q \tilde{v}_i^{m,n}, \partial_p\partial_q \tilde{v}_i^{m\pm1,n}, \partial_p\partial_q \tilde{v}_i^{m,n\pm1}\}, \tag{A1}$$

$$I_{\tau_{\rm mag,net}} = \{\bar{B}_i^{m,n}, \bar{B}_i^{m\pm1,n}, \bar{B}_i^{m,n\pm1}, \partial_p \bar{B}_i^{m,n}, \partial_p \bar{B}_i^{m\pm1,n}, \partial_p \bar{B}_i^{m,n\pm1}, \partial_p\partial_q \bar{B}_i^{m,n}, \partial_p\partial_q \bar{B}_i^{m\pm1,n}, \partial_p\partial_q \bar{B}_i^{m,n\pm1}\}, \tag{A2}$$

$$I_{\tau_{\mathrm{ind,net}}} = \{\bar{\rho}^{m,n}, \bar{\rho}^{m\pm1,n}, \bar{\rho}^{m,n\pm1}, \partial_p\bar{\rho}^{m,n}, \partial_p\bar{\rho}^{m\pm1,n}, \partial_p\bar{\rho}^{m,n\pm1}, \partial_p\partial_q\bar{\rho}^{m,n}, \partial_p\partial_q\bar{\rho}^{m\pm1,n}, \partial_p\partial_q\bar{\rho}^{m,n\pm1},$$
$$\tilde{v}_i^{m,n}, \tilde{v}_i^{m\pm1,n}, \tilde{v}_i^{m,n\pm1}, \partial_p\tilde{v}_i^{m,n}, \partial_p\tilde{v}_i^{m\pm1,n}, \partial_p\tilde{v}_i^{m,n\pm1}, \partial_p\partial_q\tilde{v}_i^{m,n}, \partial_p\partial_q\tilde{v}_i^{m\pm1,n}, \partial_p\partial_q\tilde{v}_i^{m,n\pm1},$$
$$\bar{B}_i^{m,n}, \bar{B}_i^{m\pm1,n}, \bar{B}_i^{m,n\pm1}, \partial_p\bar{B}_i^{m,n}, \partial_p\bar{B}_i^{m\pm1,n}, \partial_p\bar{B}_i^{m,n\pm1}, \partial_p\partial_q\bar{B}_i^{m,n}, \partial_p\partial_q\bar{B}_i^{m\pm1,n}, \partial_p\partial_q\bar{B}_i^{m,n\pm1}\}, \tag{A3}$$

$$I_{\tau_{\mathrm{enth,net}}} = \{\bar{\rho}^{m,n}, \bar{\rho}^{m\pm1,n}, \bar{\rho}^{m,n\pm1}, \partial_p\bar{\rho}^{m,n}, \partial_p\bar{\rho}^{m\pm1,n}, \partial_p\bar{\rho}^{m,n\pm1}, \partial_p\partial_q\bar{\rho}^{m,n}, \partial_p\partial_q\bar{\rho}^{m\pm1,n}, \partial_p\partial_q\bar{\rho}^{m,n\pm1},$$
$$\tilde{v}_i^{m,n}, \tilde{v}_i^{m\pm1,n}, \tilde{v}_i^{m,n\pm1}, \partial_p\tilde{v}_i^{m,n}, \partial_p\tilde{v}_i^{m\pm1,n}, \partial_p\tilde{v}_i^{m,n\pm1}, \partial_p\partial_q\tilde{v}_i^{m,n}, \partial_p\partial_q\tilde{v}_i^{m\pm1,n}, \partial_p\partial_q\tilde{v}_i^{m,n\pm1},$$
$$\tilde{h}^{m,n}, \tilde{h}^{m\pm1,n}, \tilde{h}^{m,n\pm1}, \partial_p\tilde{h}^{m,n}, \partial_p\tilde{h}^{m\pm1,n}, \partial_p\tilde{h}^{m,n\pm1}, \partial_p\partial_q\tilde{h}^{m,n}, \partial_p\partial_q\tilde{h}^{m\pm1,n}, \partial_p\partial_q\tilde{h}^{m,n\pm1}\}, \tag{A4}$$

where the index $i = 1, 2$ are the components of the vector, the indices m and n correspond to the discrete spatial location on the grid after filtering, and the indices $p, q = 1, 2$ represent the spatial indices along which we are taking derivatives. The $(m, n \pm 1)$ index refers to the value of the quantities in the cells located at $(m, n-1)$ and $m, n+1$, while $(m \pm 1, n)$ refers to cells designated by $(m-1, n)$ and $(m+1, n)$.

APPENDIX B: SPECTRA CALCULATION DETAILS

The spectra of the simulation $\mathcal{E}_{\mathrm{kin}}(k)$ and $\mathcal{E}_{\mathrm{mag}}(k)$ were computed in Fourier space with a 2D shifted fast Fourier transform (FFT) such that the FFT is centered at $k = 0$ under the assumption of periodic boundary conditions that were used in the simulation. We first calculated the 2D wave number $k_{2D}(k_x, k_y) = \sqrt{k_x^2 + k_y^2}$, where $k_x = 2\pi n_x/L_x$, $k_y = 2\pi n_y/L_y$, and $L_x = L_y = 1$ is the length in the x and y directions, respectively. Here $n_x \in [-N_x/2, N_x/2 - 1]$ and $n_y \in [-N_y/2, N_y/2 - 1]$ are integers, and N_x and N_y are the number of grid points in the x and y directions, respectively. We then calculated the energy spectra associated with each of these wave numbers k_x and k_y for the 2D kinetic energy and magnetic energy as

$$\mathcal{E}_{\mathrm{kin},2D}(k_x, k_y) = \frac{\widehat{\sqrt{\rho}v}_x\widehat{\sqrt{\rho}v}_x^{\,*} + \widehat{\sqrt{\rho}v}_y\widehat{\sqrt{\rho}v}_y^{\,*}}{N_x^2N_y^2}, \qquad \mathcal{E}_{\mathrm{mag},2D}(k_x, k_y) = \frac{\widehat{B_x}\widehat{B_x}^{*} + \widehat{B_y}\widehat{B_y}^{*}}{N_x^2N_y^2}, \tag{B1}$$

where $\hat{x}$ is the 2D-shifted FFT of x, rendering it a function of k_x and k_y, and x^* is the complex conjugate of x.

k_{2D} was then resampled over as $k = n\Delta k$, where $\Delta k = \sqrt{\Delta k_x^2 + \Delta k_y^2}$, $\Delta k_x = \pi/L_x$, $\Delta k_x = \pi/L_y$, and $n \in [1, N]$ is an integer. To resample, we computed $k_{\mathrm{diff}}(k, k_x, k_y) = |k - k_{2D}(k_x, k_y)|$ for every value of k_x and k_y looping over values of k. Then, for each value of k, we compute $\mathcal{E}_{\mathrm{kin}}(k)$ and $\mathcal{E}_{\mathrm{mag}}(k)$ as

$$\mathcal{E}_{\mathrm{kin}}(k) = \sum_{k_x}\sum_{k_y}\begin{cases}\mathcal{E}_{\mathrm{kin},2D}(k_x, k_y) & |k_{\mathrm{diff}}(k, k_x, k_y)| < \frac{\Delta k}{2},\\ 0 & \text{otherwise,}\end{cases} \tag{B2}$$

$$\mathcal{E}_{\mathrm{mag}}(k) = \sum_{k_x}\sum_{k_y}\begin{cases}\mathcal{E}_{\mathrm{mag},2D}(k_x, k_y) & |k_{\mathrm{diff}}(k, k_x, k_y)| < \frac{\Delta k}{2},\\ 0 & \text{otherwise.}\end{cases} \tag{B3}$$

APPENDIX C: SGS TENSORS

Here we present the plots of the SGS tensors τ_{kin}, τ_{ind}, and τ_{enth} in Figs. 12–14, respectively. Each figure provides the value of τ computed from the DNS data τ_{DNS}, the ANN model τ_{ANN}, and the gradient model τ_{grad}.

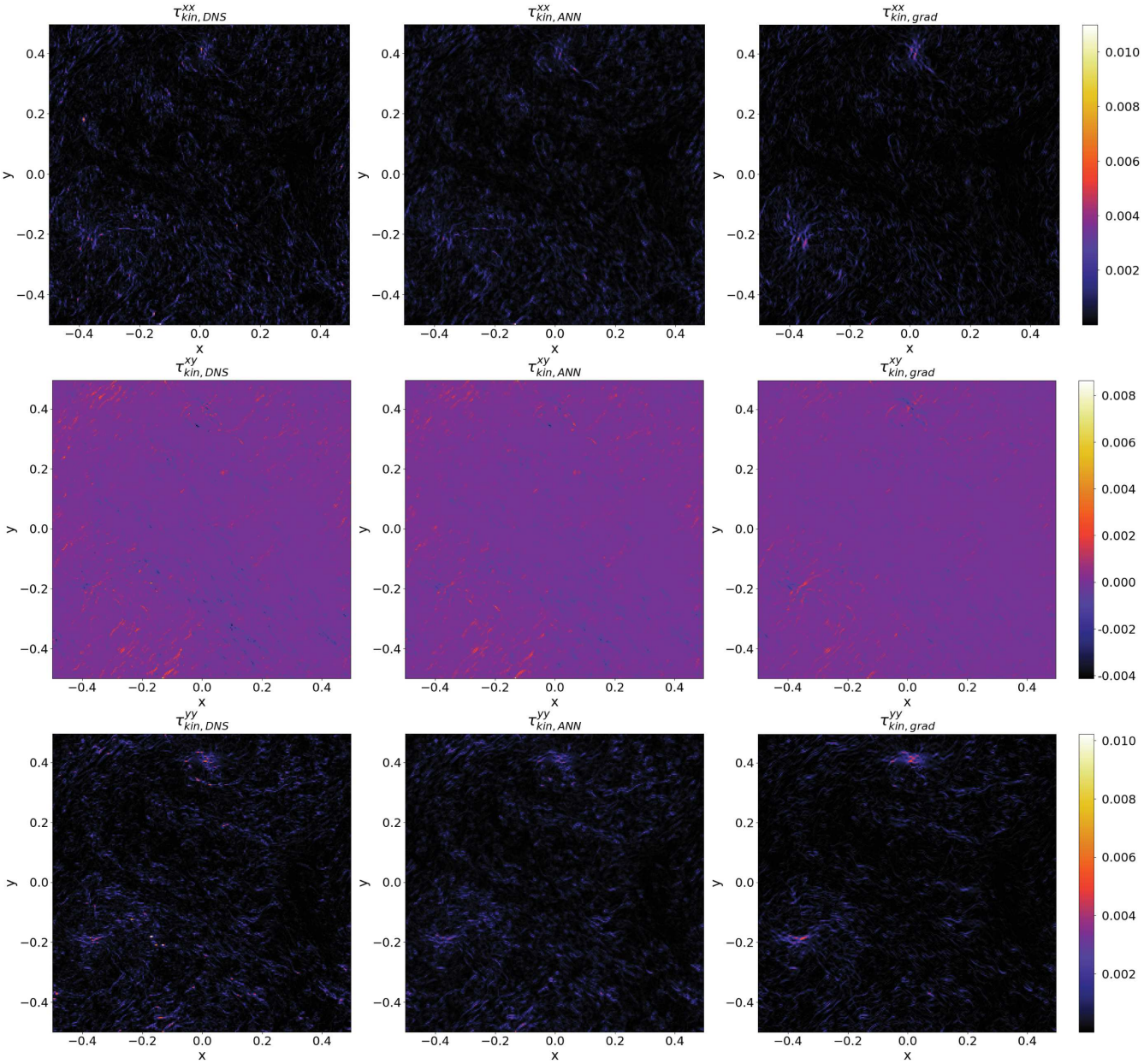

FIG. 12. The same as Fig. 8 for the SGS tensor components of $\tau_{\rm kin}$.

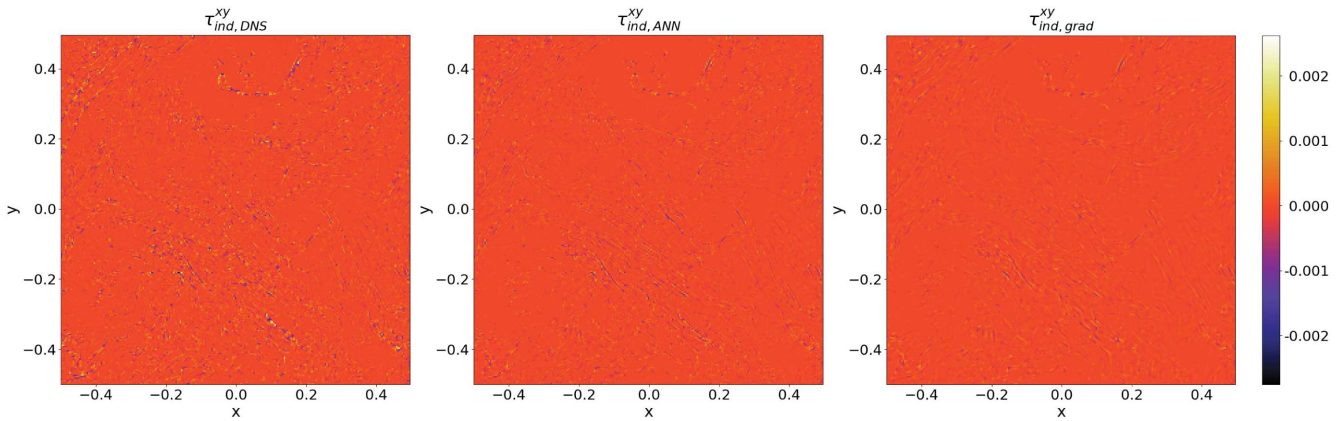

FIG. 13. The same as Fig. 8 for the SGS tensor components of $\tau_{\rm ind}$.

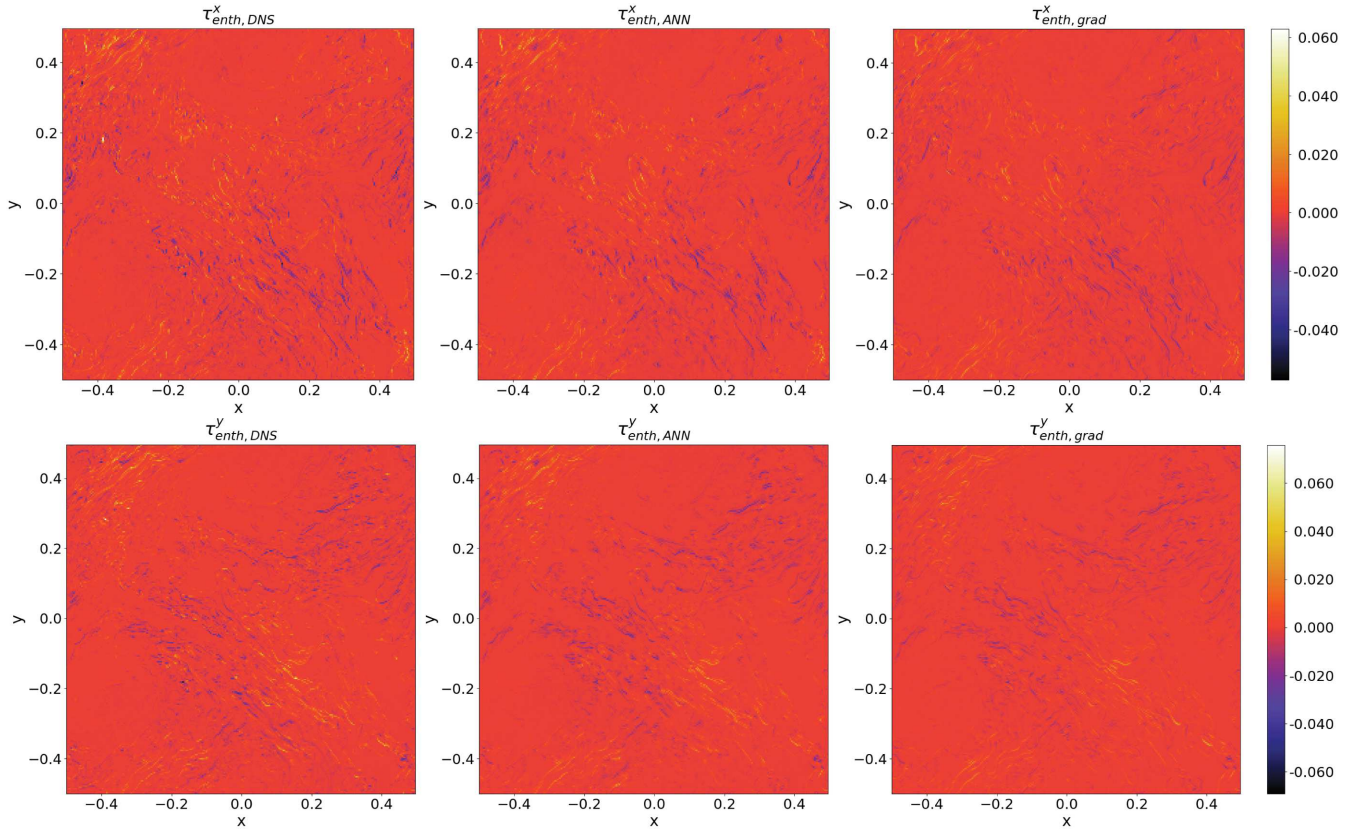

FIG. 14. The same as Fig. 8 for the SGS tensor components of τ_{enth}.

[1] W. Schmidt, Living Rev. Comput. Astrophys. **1**, 2 (2015).
[2] K. Kiuchi, P. Cerdá-Durán, K. Kyutoku, Y. Sekiguchi, and M. Shibata, Phys. Rev. D **92**, 124034 (2015).
[3] L. Baiotti and L. Rezzolla, Rep. Prog. Phys. **80**, 096901 (2017).
[4] W.-C. Mller and D. Carati, Phys. Plasmas **9**, 824 (2002).
[5] W.-C. Mller and D. Carati, Comput. Phys. Commun. **147**, 544 (2002).
[6] M. Miesch, W. Matthaeus, A. Brandenburg, A. Petrosyan, A. Pouquet, C. Cambon, F. Jenko, D. Uzdensky, J. Stone, S. Tobias, J. Toomre, and M. Velli, Space Sci. Rev. **194**, 97 (2015).
[7] P. Grete, D. G. Vlaykov, W. Schmidt, D. R. G. Schleicher, and C. Federrath, New J. Phys. **17**, 023070 (2015).
[8] P. Grete, D. G. Vlaykov, W. Schmidt, and D. R. G. Schleicher, Phys. Plasmas **23**, 062317 (2016).
[9] P. Grete, D. G. Vlaykov, W. Schmidt, and D. R. G. Schleicher, Phys. Rev. E **95**, 033206 (2017).
[10] P. Grete, B. W. O'Shea, K. Beckwith, W. Schmidt, and A. Christlieb, Phys. Plasmas **24**, 092311 (2017).
[11] M. Kessar, G. Balarac, and F. Plunian, Phys. Plasmas **23**, 102305 (2016).
[12] D. G. Vlaykov, P. Grete, W. Schmidt, and D. R. G. Schleicher, Phys. Plasmas **23**, 062316 (2016).
[13] D. Viganò, R. Aguilera-Miret, and C. Palenzuela, Phys. Rev. D **101**, 063003 (2020).
[14] F. Carrasco, D. Viganò, and C. Palenzuela, Phys. Rev. D **101**, 063003 (2020).
[15] P. Grete, Ph.D. thesis, Max-Planck-Institut für Sonnensystemforschung, 2017.
[16] A. Beresnyak and A. Lazarian, Mhd turbulence, turbulent dynamo and applications, in *Magnetic Fields in Diffuse Media*, edited by A. Lazarian, E. M. de Gouveia Dal Pino, and C. Melioli (Springer, Berlin, 2015), pp. 163–226.
[17] A. Beresnyak, Living Rev. Comput. Astrophys. **5**, 2 (2019).
[18] J. Ling, A. Kurzawski, and J. Templeton, J. Fluid Mech. **807**, 155 (2016).
[19] R. Maulik and O. San, J. Fluid Mech. **831**, 151 (2017).
[20] R. Fang, D. Sondak, P. Protopapas, and S. Succi, arXiv:1812.02241.
[21] Z. Wang, K. Luo, D. Li, J. Tan, and J. Fan, Phys. Fluids **30**, 125101 (2018).
[22] C. Xie, J. Wang, K. Li, and C. Ma, Phys. Rev. E **99**, 053113 (2019).

[23] C. Xie, K. Li, C. Ma, and J. Wang, Phys. Rev. Fluids **4**, 104605 (2019).
[24] C. Xie, J. Wang, H. Li, M. Wan, and S.Chen, AIP Adv. **10**, 015044 (2020).
[25] S. Pawar, O. San, A. Rasheed, and P. Vedula, Theor. Comput. Fluid Dyn., https://doi.org/10.1007/s00162-019-00512-z (2020).
[26] S. L. Brunton, B. R. Noack, and P. Koumoutsakos, Annu. Rev. Fluid Mech. **52**, 477 (2020).
[27] A. Vollant, G. Balarac, and C. Corre, Phys. Fluids **28**, 025114 (2016).
[28] Y. Lecun, Y. Bengio, and G. Hinton, Nature (London) **521**, 436 (2015).
[29] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. Dudley, Briefings Bioinf. **19**, 1236 (2018).
[30] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, Data Min. Knowl. Disc. **33**, 917 (2019).
[31] J. Schmidhuber, Neural Netw. **61**, 85 (2015).
[32] E. A. Huerta *et al.*, Nat. Rev. Phys. **1**, 600 (2019).
[33] A. Arbona, A. Artigues, C. Bona-Casas, J. Massó, B. Miñano, A. Rigo, and M. Trias, Comput. Phys. Commun. **184**, 2321 (2013).
[34] A. Arbona, B. Miano, A. Rigo, C. Bona, C. Palenzuela, A. Artigues, C. Bona-Casas, and J. Mass, Comput. Phys. Commun. **229**, 170 (2018).
[35] A. Dedner, F. Kemm, D. Krner, C.-D. Munz, T. Schnitzer, and M. Wesenberg, J. Comput. Phys. **175**, 645 (2002).
[36] M. Abadi *et al.*, TENSORFLOW: Large-scale machine learning on heterogeneous systems, software available from http://tensorflow.org (2015).
[37] D. P. Kingma and J. Ba, arXiv:1412.6980.
[38] S. Ghosal, AIAA J. **37**, 425 (1999).
[39] M. H. Silvis, R. A. Remmerswaal, and R. Verstappen, Phys. Fluids **29**, 015105 (2017).