A&A 629, A89 (2019)
https://doi.org/10.1051/0004-6361/201935945

Astronomy & Astrophysics

Exploring helical dynamos with machine learning: Regularized linear regression outperforms ensemble methods

Farrukh Nauman[1] and Joonas Nättilä[2]

[1] Department of Space, Earth and Environment, Chalmers University, 41296 Gothenburg, Sweden
e-mail: naumanf@chalmers.se
[2] Nordita, KTH Royal Institute of Technology and Stockholm University, Roslagstullsbacken 23, 10691 Stockholm, Sweden
e-mail: joonas.nattila@su.se

Received 24 May 2019 / Accepted 12 August 2019

ABSTRACT

We use ensemble machine learning algorithms to study the evolution of magnetic fields in magnetohydrodynamic (MHD) turbulence that is helically forced. We perform direct numerical simulations of helically forced turbulence using mean field formalism, with electromotive force (EMF) modeled both as a linear and non-linear function of the mean magnetic field and current density. The form of the EMF is determined using regularized linear regression and random forests. We also compare various analytical models to the data using Bayesian inference with Markov chain Monte Carlo (MCMC) sampling. Our results demonstrate that linear regression is largely successful at predicting the EMF and the use of more sophisticated algorithms (random forests, MCMC) do not lead to significant improvement in the fits. We conclude that the data we are looking at is effectively low dimensional and essentially linear. Finally, to encourage further exploration by the community, we provide all of our simulation data and analysis scripts as open source IPYTHON notebooks.

Key words. dynamo – magnetohydrodynamics (MHD) – turbulence

1. Introduction

Large scale magnetic fields are observed on the Earth, the Sun, galaxies. Understanding the origin and sustenance of these magnetic fields is the subject of dynamo theory. Magnetic fields observed in the galaxies are thought to have their origin in the early phases of cosmological evolution (Subramanian 2019). On relatively smaller scales, magnetic fields likely play a significant role in star (Wurster & Li 2018) and planet formation (Ercolano & Pascucci 2017). For the Sun, a long standing question has been to understand the origin of the cyclical behavior of the magnetic North and South poles (Brun & Browning 2017).

A number of outstanding questions remain about dynamo theory. Classical models of dynamo theory (Krause & Steenbeck 1967; Krause & Rädler 1980) assume that the background flow is stationary and the magnetic field is too weak to feedback on the flow. This reduces the problem of magnetic field generation to a closure problem: how does the fluctuating electromotive force (EMF) depend on the mean field? Previous work on dynamo theory has often modeled the EMF as a linear function of the mean field (see Brandenburg & Subramanian 2005, for an extensive review). This approximation is reasonable in the growth (kinematic) phase in a flow-driven magnetohydrodynamic (MHD) system where the initial magnetic energy is orders of magnitude smaller than the kinetic energy of the forced fluid. But the limitations become apparent when the magnetic fields have gained enough energy to modify the background flow through the Lorentz force. To model this backreaction of the mean magnetic field on the flow, various proposals have been put forward including models where EMF is a non-linear function of the mean field (Jepps 1975; Pouquet et al. 1976; Vainshtein & Cattaneo 1992).

Two methods have been widely used in numerical simulations of dynamos to get an explicit form for the electromotive force and both are essentially linear: the test field method (Schrinner et al. 2007) and linear regression using the least squares method (Brandenburg & Sokoloff 2002). The kinematic test field method uses a set of orthogonal "test" fields that are pre-determined and can be used to compute the various coefficients in the EMF expansion corresponding to terms linear in the large scale field and its gradients. As input, the test field requires both the small and large scale velocity field and dynamically solves (on the fly) the test field equations to compute the test field coefficients. This has the advantage of retaining dynamical information (Hubbard & Brandenburg 2009; Rheinhardt & Brandenburg 2012). Linear regression, in contrast, is mostly used in post-processing to model the "inverse" problem of determining the coefficients corresponding to different terms in the linear expansion of the EMF.

Machine learning and deep learning have revolutionized image processing, object detection, natural language processing (LeCun et al. 2015). Recently there has been a surge of interest in using machine and deep learning tools in dynamical systems and computational fluid dynamics (Hennigh 2017; Kutz 2017; Brunton & Kutz 2019). Data-driven modeling is not new to fluid dynamics; in meteorology, observational data has been used for decades to guide weather predictions (see Navon 2009, for a historical overview). Considerable increase in computational power with advances in hardware (GPUs) has allowed modern optimization methods to become increasingly tractable. The end goal is to not only build predictive data-driven models but also to understand underlying complex physics that cannot be captured by simple linear regression.

The motivation to study reduced descriptions of the full 3D MHD simulations comes primarily from the extremely large, and currently numerically intractable, magnetic Reynolds number of most astrophysical systems ($Rm \sim 10^6$–10^{12},

Brandenburg & Subramanian 2005). Reduced-order modeling of dynamical systems is currently a highly active field of research that aims to provide description of physical systems in terms of the minimal possible degrees of freedom (Kutz 2013, 2017). In this paper, we attempt to build reduced-order models for MHD dynamos by applying non-linear regression tools to the widely studied α^2 dynamo (Brandenburg & Subramanian 2005). In this context, we follow the best practices in machine learning that emphasize generalization by using train-validation-test splits in data. If one uses all the data to compute correlations (or use any other algorithm for that matter), it amounts to "descriptive" modeling. By separating the data into distinct training, validation and test sets, one enters into the realm of "predictive" modeling where the generalization power of different algorithms can be tested by first fitting data on the training set, and then testing it on the validation before making predictions on the test set. A high score on the training set but a poor score on validation set indicates poor generalization. This is in contrast to standard linear regression techniques employed in the dynamo literature where the entire data set is taken to be the training set with no validation set to test overfitting (Brandenburg & Sokoloff 2002).

We use data from Direct Numerical Simulations (DNS) of forced helical turbulence to construct a reduced model where the EMF is an unknown (linear and non-linear) function of the large scale field. This allows us to quantitatively assess the standard assumption of modeling EMF as linearly proportional to the mean magnetic field and current density. We employ three classes of modern statistical and machine learning models: (i) Linear regression, (ii) Random forests, (iii) Bayesian inference (with Markov chain Monte Carlo model minimization). Our selection of a well-studied systems like forced helical MHD turbulence supports our second main motivation: we can compare our results from machine learning tools against relatively well-known results in the literature. The insights gained from such a comparison could then help guide future work on more complex MHD turbulent systems such as MHD turbulence with differential rotation (Blackman & Brandenburg 2002; Charbonneau 2014).

We describe our data and numerical simulations in the next section. This is followed by a section that begins by explaining the fitting methods considered in this paper. We then apply these machine learning and statistical methods on the data. We discuss our results in Sect. 4 putting out work in the context of previous work and some caveats. Section 5 is conclusions. All of our analysis scripts and data are freely available as interactive IPYTHON notebooks[1].

2. Data

We describe the equations, setup and code that was used to conduct DNS of forced turbulence in magnetized flows.

2.1. Description of DNS

We use the publicly available 6th order finite difference PENCIL-CODE (Brandenburg & Dobler 2002)[2]. The MHD equations are solved in a triply periodic cubic domain of size $(2\pi)^3$:

$$\frac{\partial}{\partial t}\rho = -\nabla\cdot(\rho \boldsymbol{V}) \tag{1}$$

$$\frac{\partial}{\partial t}\boldsymbol{V} = -\frac{1}{\rho}\left[\boldsymbol{V}\cdot\nabla\cdot\boldsymbol{V} + \nabla p - 2\nabla\cdot(\nu\rho\boldsymbol{S}) - \boldsymbol{J}\times\boldsymbol{B}\right] + \boldsymbol{f} \tag{2}$$

$$\frac{\partial}{\partial t}\boldsymbol{A} = \boldsymbol{V}\times\boldsymbol{B} - \mu_0\eta\boldsymbol{J}, \tag{3}$$

[1] https://github.com/fnauman/ML_alpha2
[2] http://github.com/pencil-code

Table 1. Summary of the simulations used in this paper.

Name	$c_s/(k_1\eta)$	Rm	t_{res}	v_{rms}
R5e2	500	1.68	4.97	0.0337
R1e3	1000	4.44	9.94	0.0446
R2e3	2000	10.31	19.88	0.052
R3e3	3000	16.71	30.12	0.055
R4e3	4000	22.64	39.76	0.057
R5e3	5000	28.72	49.70	0.058
R6e3	6000	34.61	59.63	0.058
R7e3	7000	40.41	69.58	0.058
R8e3	8000	46.22	79.52	0.058
R9e3	9000	50.16	89.55	0.056
R1e4*	10 000	55.79	99.40	0.056
R15e4*	15 000	82.12	149.1	0.055

Notes. We used a resolution of 256^3 and a box size of $(2\pi)^3$ for all of our simulations. The forcing scale $k_f/k_1 = 10$ is used for all runs. Shock viscosity was used in starred (*) simulations.

where ν is the viscosity, $\boldsymbol{S} = \frac{1}{2}(V_{i,j} + V_{j,i}) - \frac{1}{3}\delta_{ij}\nabla\cdot\boldsymbol{V}$, $\boldsymbol{f}$ is the forcing function, η is the resistivity, and $\boldsymbol{J} = \mu_0^{-1}\nabla\times\boldsymbol{B}$ is the current density.

The forcing function is taken to be homogeneous and isotropic with explicit control over the fractional helicity determined by the σ parameter. It is defined as:

$$\boldsymbol{f}(t,\boldsymbol{r}) = Re\left\{N\boldsymbol{f}_{\boldsymbol{k}(t)}\exp[i\boldsymbol{k}(t)\cdot\boldsymbol{x} + i\phi(t)]\right\}, \tag{4}$$

where the wave vectors $\boldsymbol{k}(t)$ and the phase $|\phi(t)| < \pi$ are random at each time step. The wavevectors are chosen within a band to specify a range of forcing centered around $\boldsymbol{k}_f$. The normalization factor is defined such that the forcing term matches the physical dimensions of the other terms in the Navier Stokes equation, $N = f_0 c_s(|\boldsymbol{k}|c_s/\delta t)^{1/2}$, where f_0 is the forcing amplitude, c_s is the sound speed, δt is the time step. The Fourier space forcing has the form (Haugen et al. 2004):

$$\boldsymbol{f}_{\boldsymbol{k}} = \boldsymbol{R}\cdot\boldsymbol{f}_{\boldsymbol{k}}^{\rm nohel}, \quad \text{with} \quad R_{ij} = \delta_{ij} - \frac{i\sigma\epsilon_{ijk}\hat{k}_k}{\sqrt{1+\sigma^2}}, \tag{5}$$

where σ is a measure of the helicity of the forcing; for positive maximum helicity, $\sigma = 1$, and $\boldsymbol{f}_{\boldsymbol{k}}^{\rm nohel} = \boldsymbol{k}\times\hat{\boldsymbol{e}}/\sqrt{k^2 - (\boldsymbol{k}\cdot\hat{\boldsymbol{e}})^2}$.

The initial velocity and density are set to zero while the magnetic field is initialized through a vector potential with a small amplitude (10^{-3}) Gaussian noise. We define the units (Brandenburg 2001):

$$c_s = \rho_0 = \mu_0 = 1, \tag{6}$$

such that the magnetic field, $\boldsymbol{B}$, is in the units of Alfv́en speed ($c_s\sqrt{\mu_0\rho_0}$). The domain size is $L^3 = (2\pi)^3$, which leads to $k_1 = 2\pi/L = 1$. Moreover, we use the following dimensionless numbers:

$$Rm = v_{rms}/k_f\eta, \quad t_{res} = 1/k_1^2\eta, \tag{7}$$

where $k_f/k_1 = 10$ is the forcing wavemode. We provide a summary of the simulations in Table 1.

2.2. Reduced model

Numerical simulations of large scale astrophysical flows with magnetic Reynolds number well in the excess of 10^6 are prohibitive. Moreover, state-of-the-art simulations requires $O(10^6)$

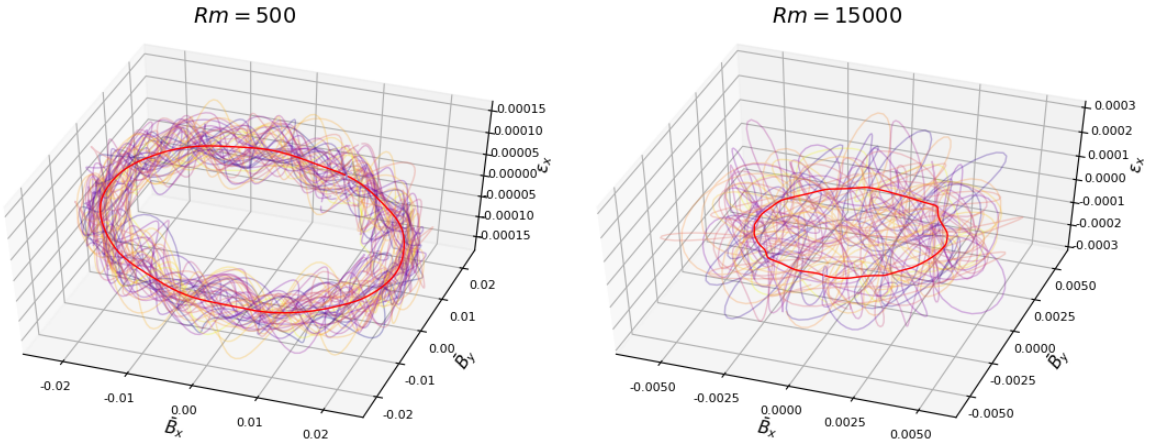

Fig. 1. 3D parametric visualization of the $\overline{B}_x(t,z)$ vs. $\overline{B}_y(t,z)$ and $\mathcal{E}_x(t,z)$ fields for different time slices as a function of z (purple curves). The thick red curves show the time averaged (beyond the kinematic regime) vertical profiles for $Rm = 500$ (*left panel*), and 1.5×10^4 (*right panel*). The small Rm simulations have a smoothly evolving circular dependencies while higher Rm cases develop secondary oscillations around the main "torus". This is consistent with the simulation becoming more "turbulent" as Reynolds number is increased.

CPU hours to compute for moderate Reynolds numbers. To make modeling such systems more tractable, it is therefore necessary to look for simpler surrogate models that can capture the most significant properties of the high fidelity simulations (DNS).

Large scale dynamo theory concerns itself with the evolution equation of the filtered induction equation,

$$\frac{\partial}{\partial t}\overline{\boldsymbol{B}} = \nabla\times(\overline{\boldsymbol{V}}\times\overline{\boldsymbol{B}}) + \nabla\times\boldsymbol{\mathcal{E}} + \eta\nabla^2\overline{\boldsymbol{B}}, \tag{8}$$

where EMF is given as $\boldsymbol{\mathcal{E}} = \overline{\boldsymbol{v}\times\boldsymbol{b}}$, η is the turbulent resistivity, and the overbar represents a filtered quantity. The filter could be ensemble, temporal or spatial depending on the problem. The first term on the right-hand side is typically ignored unless there is a strong non-constant mean velocity field such as in the case of galaxies (the mean differentially rotating flow for the galactic dynamo is considered in Shukurov et al. 2006). The dynamics of the velocity field (Eq. (2)) are ignored based on the assumption that the forcing term is the dominant term and it causes the velocity field to be in a steady state. This assumption is only reasonable in the "kinematic" regime where the magnetic fields are too weak to modify the velocity field dynamically.

In keeping with previous work that has used horizontal averaging as the filter in Eq. (8) (Brandenburg 2001; Brandenburg & Subramanian 2005), our data has been averaged over the whole xy plane:

$$\overline{\boldsymbol{B}} = \frac{1}{L_xL_y}\int_0^{L_y}\int_0^{L_x}\boldsymbol{B}\mathrm{d}x\mathrm{d}y. \tag{9}$$

This averaging scheme has the interesting property that $\nabla\times(\overline{\boldsymbol{V}}\times\overline{\boldsymbol{B}}) \sim 0$ since $\overline{B}_z = 0$ (due to incompressibility and periodic boundary conditions). We only consider the fields (let Q be an arbitrary field) reduced to 1D by furthermore averaging them in two different ways:

1. z averaged: $\left\langle\overline{Q}\right\rangle(t) = 1/(z_2-z_1)\int_{z_1}^{z_2}\overline{Q}\mathrm{d}z$ where z_1, z_2 are chosen such that they do not cover the entire z-domain since that will correspond to volume averaged quantities that are zero.
2. t averaged: $\left[\overline{Q}\right](z) = 1/(t_2-t_1)\int_{t_1}^{t_2}\overline{Q}\mathrm{d}t$ where t_1, t_2 are chosen to correspond to either the kinematic or saturation regime for different runs.

This leaves two independent magnetic fields and corresponding EMF components. Analytical models that use $\boldsymbol{\mathcal{E}} = \alpha\overline{\boldsymbol{B}} - \eta_t\overline{\boldsymbol{J}}$ are termed α^2 dynamos and have been extensively studied in the literature. Here η_t represents the turbulent resistivity.

In Fig. 1, we show a three-dimensional phase-space visualization of $\mathcal{E}_x$ vs. $\overline{B}_x$ vs. $\overline{B}_y$ that shows the relation between the EMF and the mean magnetic fields (situation is the same when considering $\mathcal{E}_y$ too). The circle in $\overline{B}_x - \overline{B}_y$ plane represents the fact that the total magnetic energy $\overline{B}_x^2 + \overline{B}_y^2 \sim$ constant. The slight tilt with respect to the EMF (i.e., towards z direction) implies a linear correlation with the EMF. The increasing disorder with increasing Rm is characteristic of highly turbulent systems.

3. Model fits

We describe the 1D linear and non-linear regression for temporal and vertical profiles in the following subsections. In the following, we will refer to the magnetic fields as both "fields" and "features", and EMF as "target" interchangeably. For most of the following, we will focus on $\mathcal{E}_x$ but similar results hold for $\mathcal{E}_y$ since we are looking at isotropically forced MHD turbulence with no background field.

3.1. Description of algorithms considered in this paper

Ordinary Least Squares (OLS) assumes that the data is linear, features/terms are independent, variance for each point is roughly constant (homoscedasticity), features/terms do not interact with one another. Moreover, causal inference based on linear regression can be misleading (Bollen & Pearl 2013), a problem shared by all curve fitting methods. We should highlight the distinction between linearity of the regression as opposed to linearity of the features. For example, one can construct a non-linear basis (example: $x, x^2, x^3, \ldots$ or $\sin x, \cos x, \ldots$) and do linear regression with this basis. Linear regression on non-linear data can potentially represent non-linear data well assuming that the non-linear basis terms are independent and have low variance. Constructing the correct interacting/non-linear terms requires domain expertise. For this reason, it is important to also consider non-linear regression methods such as random forests that are robust to noise and are capable of modeling interactions implicitly.

3.1.1. LASSO

Most real data have considerable variance that can lead to misleading fits from linear regression. Regularized linear regression aims to construct models that are more robust to outliers. Two of the most common regularization schemes for linear regression are (Tibshirani 1996; Hastie et al. 2015): (i) LASSO (Least Absolute Shrinkage and Selection Operator; also known simply as L1 norm), (ii) Ridge (also known as L2 norm). As opposed to ridge, LASSO has the advantage of doing feature selection by reducing coefficients of insignificant features. We will use LASSO regression in this work.

Intuitively LASSO reduces variance at the cost of introducing more bias. LASSO works better than ridge regression if the data is low dimensional (or the coefficient matrix is sparse). For high dimensional complex data, neither ridge nor LASSO do well. Because of this, the "Bet on sparsity" principle (Hastie et al. 2009) suggests trying LASSO for all datasets. Formally, LASSO optimizes for:

$$\mathrm{argmin}_w \left[\|y - Xw\|_2^2 + \alpha \|w\|_1 \right], \tag{10}$$

where y represents the target vector (in our case, $\mathcal{E}$), $\boldsymbol{X}$ is the feature matrix ($B_x, B_y, \ldots$ are the columns), w is coefficient vector and α is a parameter with a range between 0 and 1. The subscripts "1" and "2" represent L1 ($\sum_n \|w\|_1 = |w_1| + |w_2| + \ldots + |w_n|$) and L2 ($\sum_n \|w\|_2 = \sqrt{w_1^2 + w_2^2 + \ldots + w_n^2}$) norms, respectively. A higher α penalizes outliers more strongly and shrinks coefficients of unimportant features to zero while a lower α is similar to linear regression with no regularization (OLS).

3.1.2. Random forests

Random forests belong to the class of ensemble machine learning algorithms (Breiman 2001a; Hastie et al. 2009). A random forest consists of a large number of decision trees that each take a random subset of features with a bootstrapped sample of the data (Raschka 2018). This helps in getting rid of strong correlations among different trees but some correlations might still remain. Random forests are one of the most popular and widely used machine learning algorithms because they require little hyperparameter tuning and can handle large noisy data sets. From a computational point of view, each decision tree in the ensemble is independent and can thus be easily processed in parallel. Moreover, because of the weighted average over the ensemble, random forests are robust to strong variance in the data but a few deep decision trees can lead to strong bias.

Ensembles of decision trees (random forests and gradient boosting) are non-parametric models meaning they do not require information about the underlying data distribution. In other words, unlike in linear regression where the shape of the function that connects the independent variables to the dependent variable is fixed, the functional form is not known in random forests and is instead determined through training. This offers a different perspective on modeling, termed "algorithmic" modeling as opposed to "data" modeling (Breiman 2001b). The non-parametric nature of random forests and gradient boosting make them applicable to a wide class of problems since in most real-world situations, data distribution is not known a-priori. However, this comes at the cost of interpretability making ensembles of decision trees harder to understand than linear regression (Strobl et al. 2008).

Feature importance and selection: As opposed to linear regression, random forests do not directly measure the coefficient of the features. A single decision tree is highly interpretable but a whole ensemble of decision trees is harder to interpret (Breiman 2001b). Random forests offer some insight through feature importances (Hastie et al. 2009). We use the "mean decrease impurity" method implemented in PYTHON library SCIKIT-LEARN (Pedregosa et al. 2011; Breiman et al. 1984; Louppe 2014). With this method, the feature importance corresponds to the frequency of a particular feature appearing across different trees combined with the reduction of error corresponding to that particular feature, weighted by the fraction of samples in that particular node. One popular alternative is to randomly permute the data for a particular feature and see whether it has an effect on the mean squared error (or information gain/gini importance for classification tasks). If the mean squared error increases or remains the same, it implies that the particular feature is not important. This is termed as "permutation" importance (Breiman 2001a; Strobl et al. 2008).

3.1.3. Bayesian inference

Bayesian inference is a powerful statistical framework that presents an alternative way to perform the model parameter estimation. The main strengths of Bayesian-based approaches are their ability to quantify the uncertainty of the model and the possibility to incorporate previous a-priori knowledge into the estimation. The downside, when compared to many conventional machine learning methods, is that the model needs to be known before the fit.

Let us quickly summarize the idea behind Bayesian inference. In general, let us present our model with $\mathcal{M}$ and our empirical (simulation) data with $\mathcal{D}$. We are then interested in how well can our model describe the data, or quantitatively what is the probability that our model is valid given the data, $\Pr(\mathcal{M}|\mathcal{D})$. By directly comparing our model and the data, what we actually get is, however, the probability of our model given the data, known as the likelihood $\mathcal{L} = \Pr(\mathcal{D}|\mathcal{M})$. Our original question of the validity of our model, can be answered by applying the Bayes' theorem presented as (see e.g., Grinstead & Snell 1997)

$$\Pr(\mathcal{M}|\mathcal{D}) = \frac{\Pr(\mathcal{D}|\mathcal{M})\Pr(\mathcal{M})}{\Pr(\mathcal{D})}, \tag{11}$$

where $\Pr(\mathcal{M})$ is our prior probability of the model and $\Pr(\mathcal{D}) = \int \Pr(\mathcal{D}|\mathcal{M}) \mathrm{d}\mathcal{M}$ is the prior probability. Therefore, because $\mathcal{L}$ can be computed and $\Pr(\mathcal{M})$ is known a-priori, we can use the Bayes' theorem to quantitatively asses the validity of, for example, our α^2 dynamo models given the simulation data results.

In practice, not only are we interested in the validity of the model, but also what are the parameters Θ of the model, $\mathcal{M}(\Theta)$, that best describe the data. In Bayesian inference, these model parameters are determined using a marginal estimation where the resulting posterior parameter for model parameter Θ_j are obtained by integrating over the probability of all the other model parameters, Θ_i ($i \neq j$). Then, this (one-dimensional) distribution represents the probability that the jth model parameter, Θ_j, will take a particular value given our data $\mathcal{D}$.

Here we solve the Eq. (11) (and therefore obtain also posterior distributions for our model parameters) using Monte Carlo Markov chain (MCMC) integration techniques. To perform the MCMC fit we use the publicly available EMCEE library (Foreman-Mackey et al. 2013). We employ the affine-invariant stretch-move ensemble sampler with typically $3N_{\mathrm{param}}$ number of walkers (members of ensemble), where N_{param} is the number of fit parameters. No prior knowledge is incorporated into the fits and so we use uniform priors for all of our model parameters.

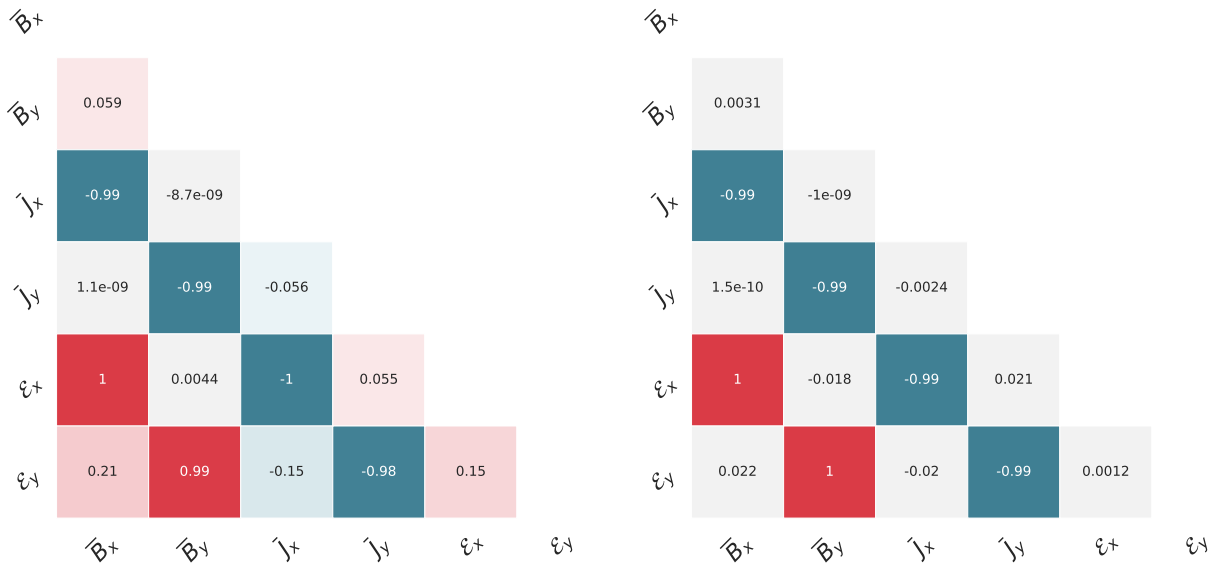

Fig. 2. Heat map of correlation coefficients between the different fields: $Rm = 10^3$ (*left panel*), $Rm = 1.5 \times 10^4$ (*right panel*). This analysis implies that the $\mathcal{E}_x$ has a strong correlation with $\overline{B}_x, -\overline{J}_x$ while $\mathcal{E}_y$ has a strong correlation with $\overline{B}_y, -\overline{J}_y$. Moreover, mean fields $(\overline{B}_x, \overline{B}_y)$ are not linearly independent of current densities $(\overline{J}_x, \overline{J}_y)$.

3.2. Vertical profiles

In this subsection, we apply various methods on time averaged data that only has vertical dependence (256 grid points).

In Fig. 2, we show the linear (Pearson) correlation coefficients among the various fields. The correlation between the current densities and the magnetic fields is nearly 1 indicating they are linearly dependent. Strong linear correlation does not rule out the physical importance of a variable – it only implies that from a curve fitting point of view, it has redundant information. Since we are using helical forcing leading to a helical state, $\overline{\boldsymbol{J}} = \nabla \times \overline{\boldsymbol{B}} \sim \overline{\boldsymbol{B}}$. At high Rm, one might expect that the power in the velocity and magnetic field spectrum will be distributed across multiple modes implying weaker correlations. However, Fig. 2 (right panel) indicates the correlation between $\overline{\boldsymbol{B}}$ and $\overline{\boldsymbol{J}}$ is still perfect. This means that we can eliminate current density as an independent variable. See, however, the work Tilgner & Brandenburg (2008) that suggests that the tensorial form of α_{ij}, η_{ij} in the non-linear regime depends on $\overline{B}_i \overline{B}_j / \overline{\boldsymbol{B}}^2$. In keeping with previous work, for linear basis we do consider the current density as an independent variable but eliminate it for the non-linear (polynomial) basis where its inclusion will lead to several redundant terms.

We consider two sets of basis in this subsection:

1. Linear: $\overline{B}_x, \overline{B}_y, \overline{J}_x, \overline{J}_y$.
2. Polynomial of $O(3)$: $\overline{B}_x, \overline{B}_y, \overline{B}_x\overline{B}_y, \overline{B}^2, \overline{B}^2\overline{B}_x, \overline{B}^2\overline{B}_y$.

We note that for kinetically forced simulations, $v^2 \sim$ constant, which means that the total energy conservation equation yields, $\overline{B}_x^2 + \overline{B}_y^2 = \overline{B}^2 \sim$ constant in the saturated state. This makes $\overline{B}_x^2$ and $\overline{B}_y^2$ linearly dependent terms and thus we do not consider them separately. Another motivation to use $\overline{B}^2$ is because work on dynamo quenching often considers $\alpha \sim 1/(1 + \overline{B}^2/B_{\rm eq}^2)$, which when expanded for the $\boldsymbol{\mathcal{E}}$ leads to cubic terms of the form $\overline{B}^2\overline{B}_i$ $(i = x, y)$. We will be using the publicly available PYTHON library SCIKIT-LEARN (see Guido & Müller 2017; Géron 2017, for introductions) for modeling the data.

Previous work on dynamo theory has often relied on linear regression methods (Brandenburg & Sokoloff 2002; Brandenburg 2018) with some effort to incorporate non-linear effects through magnetic helicity conservation (Pouquet et al. 1976; Blackman 2015; see also non-linear test field method Rheinhardt & Brandenburg 2010). In this section, we use linear regression with a regularization term for both the linear and polynomial basis discussed previously.

In Fig. 3, we show the fits using random forests regression and LASSO. Because our data is small (only 256 points), we use two-fold cross validation to determine various hyperparameters of LASSO and random forests. We found that bootstrapping leads to spurious results so we turned it off. A similar problem was found with random train/test splits: the spatial structure of the fields is like one full sinusoidal wave and randomly picking a fraction of the points leads to strange patterns. Moreover, while deep trees can lead to overfitting, in our case we found that a maximum depth of 8 was optimal through 2-fold cross validation. The end results are still not as good as LASSO perhaps owing to the fact that the training set is only 204 entries (80% of the data) – this is too low for an algorithm like random forest to really excel. That is the key lesson we have learnt while modeling this dataset throughout: "small" datasets mean simpler algorithms will outperform more sophisticated algorithms like random forests.

In Table 2, we compare the coefficients from OLS, LASSO, randomized LASSO (stability), Recursive Feature Elimination (RFE) from OLS and RFE_{rf} from random forests, Mutual Information Criterion (MIC), Pearson correlation (Corr) and the average of all these predictions in the last column. The meaning of these numbers is different. For OLS and LASSO, the numbers corresponding to the fields $\overline{B}_x$ represent coefficients from linear regression and can be negative or positive. For random forests, the numbers represent the relative feature importance based on "mean decrease impurity" that is only positive. These feature importance numbers are scaled to sum to unity. Stability selection is a general term that refers to checking robustness of model predictions by testing it on different random (bootstrapped) subsets of data with different hyperparameter values. In our

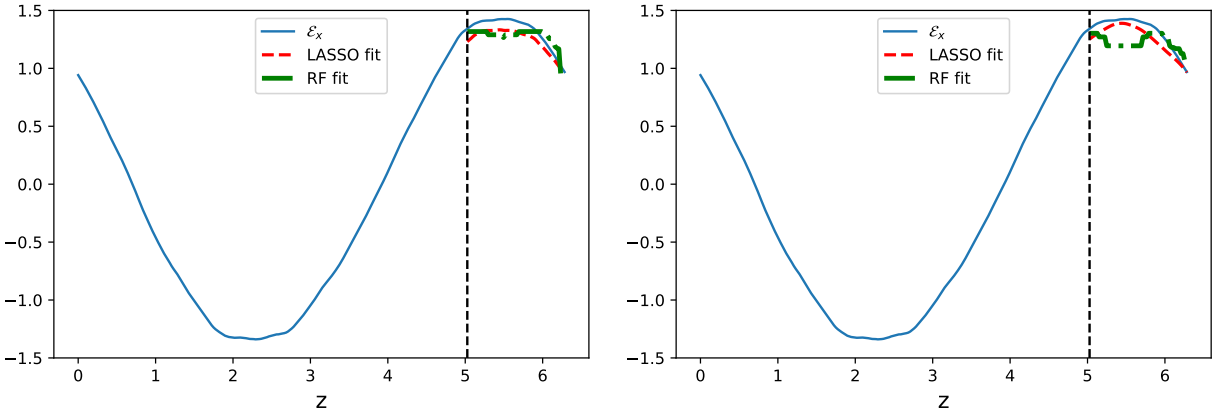

Fig. 3. Fits for linear basis (*left panel*) and polynomial basis (*right panel*) for random forests and LASSO. Note that the horizontal lines seen in the fits are there because of decision tree regression: decision trees set one value of the y-axis for a given range of x-axis leading to a step-function like fit. In both cases, LASSO seems to do a better job at capturing the curved shape of the vertical profiles. The black vertical dashed line presents the train-test split (80% training data).

Table 2. Feature importances using various machine learning models and statistics.

Feature	OLS	Lasso	Stability	RanFor	$\mathrm{RFE_{OLS}}$	$\mathrm{RFE_{rf}}$	MIC	Corr	Mean
B^2	0.0076	0.0048	0.0	0.0049	0.2	0.6	0.0000	0.1019	0.1149
B^2B_x	0.6345	0.2577	0.0	1.0000	0.8	1.0	1.0000	0.9972	0.8362
B^2B_y	0.0871	0.0240	0.0	0.0000	0.6	0.0	0.3979	0.1210	0.1538
B_x	1.0000	1.0000	1.0	0.4404	1.0	0.8	1.0000	0.9982	0.7793
B_y	0.0818	0.0000	0.0	0.0000	0.4	0.4	0.4048	0.1179	0.1756
B_xB_y	0.0000	0.0145	0.0	0.0001	0.0	0.2	0.2089	0.1366	0.0700

Notes. All the numbers are normalized such that they are positive and lie between 0 and 1 for the sake of comparison. Stability column refers to randomized LASSO where the LASSO shrinkage coefficient is randomly varied for different features and the feature that is most robust to this variation survives. RFE stands for Recursive Feature Elimination where a model is trained with all features and the top ranking feature is given the most significance, while the bottom ranking one gets the smallest score. We applied this to both OLS and random forests, and reverse sorted the entries to give the highest score to top ranking feature. MIC stands for Maximal Information Coefficient and computes a normalized measure of the mutual information between two variables scaled between 0 and 1. It gives a quantitative measure of the question: how much information about some variable Y can be obtained through some variable X? MIC is capable of capturing non-linear relationships that Pearson correlation (Corr) cannot. In the last column we take the mean over all models that represents the synthesis of several models (linear, non-linear) to show that $\overline{B}_x$ and $\overline{B}^2\overline{B}_x$ are the two strongest features.

particular application, we are using randomized LASSO that randomly varies the LASSO coefficient (Meinshausen & Bühlmann 2010)[3].

RFE is a model-agnostic (wrapper) feature selection method that starts with a number of features and removes the most important features recursively. This is an example of "backward" feature selection (Guyon & Elisseeff 2003; Hastie et al. 2009) and we used it with OLS and random forests in this case. Maximal Information Coefficient (MIC) is a measure of mutual information between two (random) variables and is capable of accounting for non-linear relationships that the Pearson correlation coefficient cannot (example: correlation between x and x^2 will be nearly zero, but MIC will yield an answer close to one). We scaled the whole set of features X_{train} using standard scaling before calculating coefficients/feature importances. Moreover, because LASSO, OLS, Corr returned negative as well as positive values and RFE returned values ranging from 1 to 6, we used minmax scaling to re-scale all values between 0−1.

Ensembles of machine learning models can lead to improved accuracy (Wolpert 1992). Here we take the simplest possible approach: we take the unweighted average of various predictors to compute the overall score for each feature. While stability selection shows that $\overline{B}_x$ is the dominant feature, the mean of all predictors gives $\overline{B}^2\overline{B}_x$ a slightly higher score. This could be because both $\overline{B}_x$ and $\overline{B}^2\overline{B}_x$ have the same sinusoidal form.

What do these results mean? Because we are dealing with low-dimensional data, constructing higher order terms from linear terms is not beneficial. LASSO is particularly well suited to identify important features in a low-dimensional model. It is, therefore, no surprise that the most important feature that (both randomized and ordinary) LASSO is picking up is the linear term $\overline{B}_x$. Random forests, on the other hand, pick up $\overline{B}^2\overline{B}_x$ as the most significant feature and this might explain the poor prediction based on random forests seen in Fig. 3. Random forests work best for large datasets; for small data that is really low-dimensional (that is nearly linear), random forests do not perform well. Therefore, the results of Table 2 are consistent with Fig. 3.

3.3. Time series analysis: Vertically averaged data

In contrast to the previous section, the magnetic fields and the EMF show exponential growth in time. To model this behavior,

[3] https://blog.datadive.net/selecting-good-features-part-iv-stability-selection-rfe-and-everything-side-by-side/; code: https://github.com/scikit-learn-contrib/stability-selection

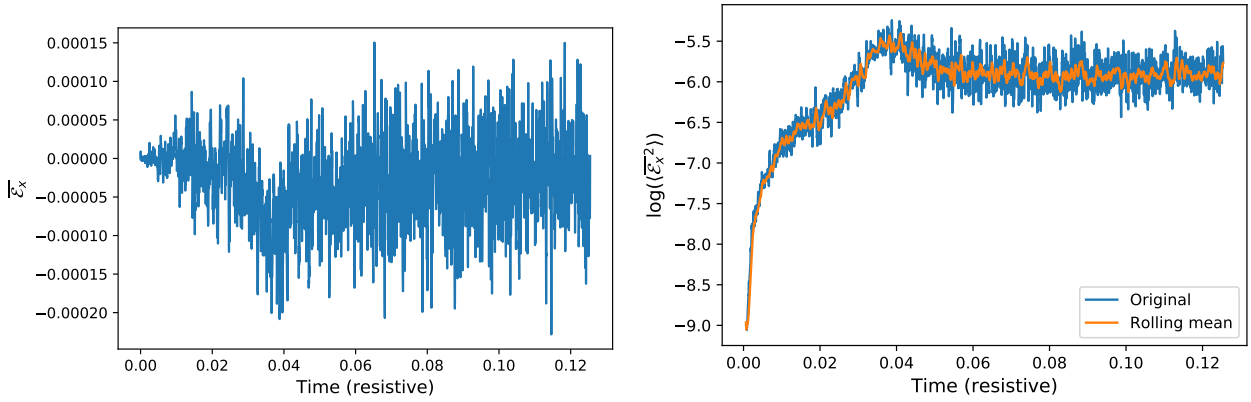

Fig. 4. EMF evolution for $Rm = 1.5 \times 10^4$. *Left*: $\mathcal{E}_x$ component is shown without any pre-processing ($z = \pi$). *Right*: here we show $\log\{\langle\mathcal{E}_x^2\rangle\}$, that is the squared EMF averaged over z. Moreover, we use a moving average with window size 20 for the orange line to smooth out the fluctuations.

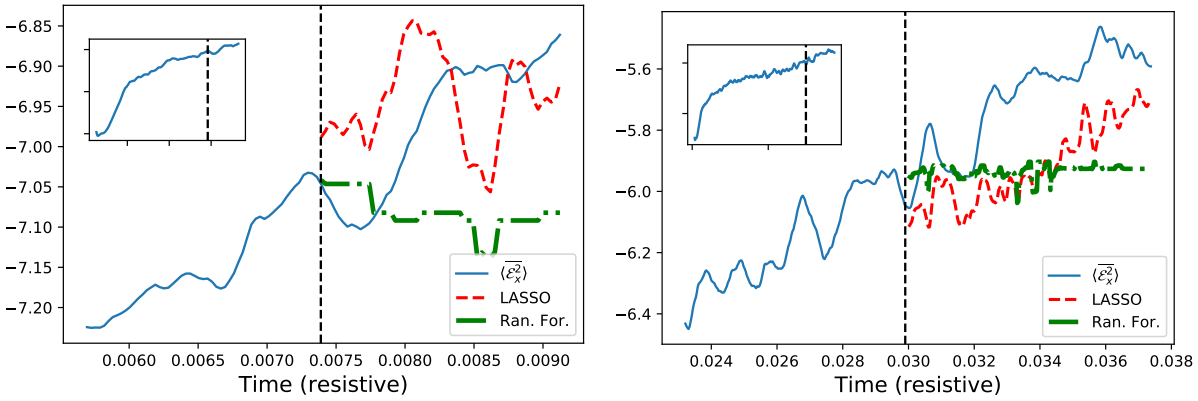

Fig. 5. The fits for $\mathcal{E}_x^2$ time series using LASSO and random forests for $Rm = 1.5 \times 10^4$. Just like the fits for spatial data in Fig. 3, we find that LASSO does considerably better than random forests, the latter of which again has a step like shape. *Left*: we show the time series only up to the kinematic phase (∼0.01 resistive times) with the inset describing the train/test split. *Right*: time series is between kinematic and saturation phases. LASSO seems to capture the shape of the curve but is offset. In both cases, random forest prediction returns a nearly horizontal line, characteristic of decision tree regression. The black vertical dashed line presents the train-test split (80% training data).

we found it convenient to do pre-processing where we took the logarithm of squared vertically averaged fields. Because of this choice, we do not consider polynomial expansions like we did before and instead focus on either the linear $\boldsymbol{\mathcal{E}} \propto \overline{\boldsymbol{B}}$ form, or the quenched form $\boldsymbol{\mathcal{E}} \propto \overline{\boldsymbol{B}}/(1 + \overline{B}^2/B_{eq}^2)$.

3.3.1. Preprocessing

For the vertically averaged data set, we find that the fluctuations in the EMF are quite significant (see Fig. 4). Such large fluctuations make any kind of statistical analysis quite prohibitive. In this section, we therefore decided to use the log of $\mathcal{E}_x^2$ averaged over the z direction. Moreover, to reduce the fluctuations we used a moving window of size 20 (right panel of Fig. 4). This allows for cleaner fits.

3.3.2. LASSO and random forest

In Fig. 5, we show fits for the $\mathcal{E}_x^2$ time series using LASSO and random forests. Time series data is particularly difficult to fit in the presence of fluctuations and trend as in the figure of $\mathcal{E}_x^2$. In the kinematic phase (left panel), the random forest fit is not considerably different from LASSO fit whereas the random forest fit does poorly when the same time series is extended up to the saturation phase (right panel). This is a relatively common issue with random forests that they do not perform well when there is a "trend" present in the series. Just like the spatial profiles in last subsection, the LASSO outperforms random forests.

3.3.3. Bayesian inference

In this section we focus on reconstructing square of the "true" EMF field $\boldsymbol{\mathcal{E}}^2$ as a function of time t. As one of the simplest forms possible, we construct the reconstructed EMF, $\boldsymbol{\mathcal{E}}_{\rm R}(z;t)^2 = \mathcal{E}_{x,{\rm R}}(z;t)^2 + \mathcal{E}_{y,{\rm R}}(z;t)^2$, from the corresponding magnetic field components as

$$\mathcal{E}_{x,{\rm R}}(z;t) = \frac{\alpha}{1 + \overline{B}^2(z;t)/B_{\rm eq}^2}\overline{B}_x(z;t) \tag{12}$$

$$\mathcal{E}_{y,{\rm R}}(z;t) = \frac{\alpha}{1 + \overline{B}^2(z;t)/B_{\rm eq}^2}\overline{B}_y(z;t), \tag{13}$$

where $\mathcal{E}_{x,{\rm R}}^2$ and $\mathcal{E}_{y,{\rm R}}^2$ are the reconstructed x and y components of the EMF field. Here α and $B_{\rm eq}$ are our model parameters to optimize. This EMF form has the nice property that in the limit $B_{\rm eq} \to \infty$ it reduces to the linear model.

After this, we are left with the freedom to define our likelihood or score-function that describes the quality of the reconstruction. Here we define it via a simple residual function $\mathcal{R}$,

$$\mathcal{R} = \frac{\boldsymbol{\mathcal{E}}_R}{\boldsymbol{\mathcal{E}}} + \frac{\boldsymbol{\mathcal{E}}}{\boldsymbol{\mathcal{E}}_R} - 2, \tag{14}$$

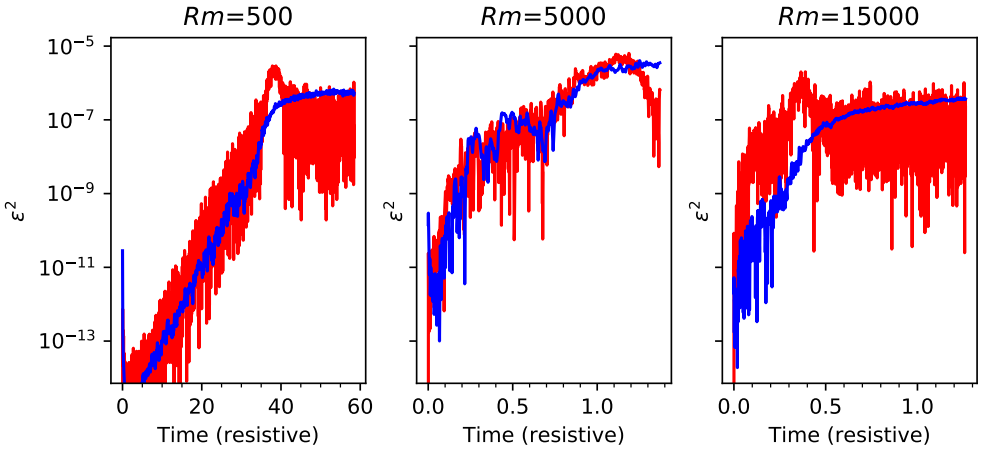

Fig. 6. MCMC fits for the reconstructed $\mathcal{E}^2$ fields. Three panel shows maximum likelihood Monte Carlo realizations of the fits (blue line) against the true $\mathcal{E}^2$ curve (red line) for $Rm = 500$, 5×10^3, and 1.5×10^4.

Table 3. Summary of the advantages and disadvantage of different models considered in this work.

Property	OLS	LASSO	RF	Bayesian
Non-linear	No	No	Yes	Yes
Interpretable	Yes	Yes	Yes [(a)]	Yes
Suitable for all data sizes?	Yes	Yes	Mid-sized to big data	Small to mid-sized data
Interactions	No	No [(b)]	Yes	Yes
Prediction of trends	No	No	No	Yes
Poor signal to noise ratio	No	No	Yes	Yes

Notes. [(a)]Feature importances and partial dependence plots can help interpret random forests. [(b)]Including cross terms like $\overline{B}_x\overline{B}_y$ can help model interactions but it still assumes each feature is linearly independent.

that penalizes the fit from over- and underestimating the reconstructed EMF. Note that other forms of the residual functions are also possible. For discrete simulation data results (defined on an array) the likelihood then simplifies to $\mathcal{L}_q = \sum_k |\mathcal{R}_k|$, where the summation index k is taken to be over the time dimension or the z direction, depending on the type of fit we are considering. Finally, when employing the MCMC fit note that we are actually minimizing $\log \sum_q \mathcal{L}_q$, as is typically done, to get a more shallow and slowly varying score-function.

Examples of the fit results are shown in Fig. 6 for three different Rm values of 500, 5×10^3, and 1.5×10^4 depicting the maximum a posteriori model. The (quenched) EMF model does reproduce the qualitative behavior of the saturating $\overline{\boldsymbol{B}}$ field but especially for higher Rm the fits become more worse. Most importantly, we have found that constraints on the saturation field strength $B_{\rm eq}$ are always very loose. Results without using the quenching form also lead to similar fits. This gives an indirect support for the previous machine learning results: The data strongly favors isotropic α with a simplified model of $\boldsymbol{\mathcal{E}} = \alpha\overline{\boldsymbol{B}}$. The meaning of this "$\alpha$" could be confusing since we have already eliminated the mean current density $\overline{\boldsymbol{J}}$ as an independent variable. The α coefficient represents the saturated value where the growth term roughly balances the dissipation term, leading to the effective expression: $\alpha \sim k_1(\eta + \eta_t)$.

3.4. Results summary

For both the vertical and temporal profiles, various models suggest that $\boldsymbol{\mathcal{E}} \propto \overline{\boldsymbol{B}}$ is a good description of the data. But if $\mathrm{d}\overline{\boldsymbol{B}}/\mathrm{d}t \propto \alpha\overline{\boldsymbol{B}}$, this will lead to exponential growth without any saturation. Is there an inconsistency here? In the induction equation, the fields at time "t" are related to the fields at earlier times. The machine learning models we use here are not capable of storing long term memory. The fits shown in this paper are comparing $\boldsymbol{\mathcal{E}}$ and $\overline{\boldsymbol{B}}$ in some nearest neighbor sense and because both $\boldsymbol{\mathcal{E}}$ and $\overline{\boldsymbol{B}}$ are growing exponentially before saturating at about the same time, they appear to be linearly correlated. Capturing the dynamical information requires more sophisticated algorithms that we leave for future work.

4. Discussion

4.1. Caveats

We summarize the properties of the different machine learning algorithms considered in this work in Table 3. Linear regression (both OLS and LASSO) are by definition capable of dealing with linear data only. However, they can handle some non-linear properties using a non-linear basis (for example, polynomial regression). The biggest advantage of linear models is the interpretability: the sensitivity of the target variable (EMF) with respect to the features $(\overline{B}_x, \overline{B}_y)$ is as easy as determining the coefficients of these terms. Linear models are also good at dealing with data of all sizes but matrix inverses are expensive for large data. Linear models might also be able to model trends in data if the features they consider have the same trend as the target variable.

Random forests are based on ensembles of decision trees. While random forests are quite robust to outliers and missing data, the feature importances computed from random forests can be misleading (Strobl et al. 2008). Moreover, random forests are not as interpretable as linear regression or single decision trees. A typical random forest fit can involve 10−100 trees implying that interpretation is difficult. But computation of feature importances and partial dependence plots help in making sense of random forest results (Breiman 2001b; Hastie et al. 2009). Random

forests typically bootstrap data and use a random subset of features for each decision tree to determine which one leads to the lowest error. These two sources of randomness imply that random forests are best suited for intermediate to big data. Indeed in our "small data" case, we find that bootstrapping had to be turned off and the best fit models used less than 10 decision trees.

Bayesian methods offer an interesting complementary toolkit to various machine learning techniques. They also enable us to properly take the noisy nature of DNS into account. Here the big caveat is, that the model(s) needs to be known beforehand. This, however, further arguments in favor of our hybrid approach applied in this paper. We have used machine learning techniques to detect important features and then applied our physical knowledge to build valid models out from those. Biggest caveat here is that Bayesian model optimization becomes more and more computationally demanding with multidimensional (and often multimodal) data. This puts a limitation on the real-life usability of the method when dealing with increasingly complex data.

The data considered in this paper is either spatially or temporally averaged reducing to $O(100-1000)$ entries. Big data typically refers to the case where the observations/rows are $\geq 10^6$. "Small" data faces problems of overfitting and is more likely to be sensitive to outliers. Because of these issues, simpler models like linear regression with regularization tend to do well. It is, therefore, no surprise that LASSO has outperformed random forests. For larger datasets with irregular non-linear patterns, random forests and gradient boosting tend to do much better (Olson et al. 2017).

4.2. Comparison with previous work

We used the simplest model: isotropic with no non-local effects in either space and/or time. While earlier work has focused primarily on linear models of dynamos where $\boldsymbol{\mathcal{E}} \sim \overline{\boldsymbol{B}}$ using linear regression and the test field method (Brandenburg & Subramanian 2005), non-linear models have also been considered (Rheinhardt & Brandenburg 2010). We assumed isotropy, which could be a misleading assumption in the presence of shear (see Brandenburg & Sokoloff 2002; Karak et al. 2014). Furthermore, in our model, we did not incorporate non-local effects that were considered by Brandenburg & Sokoloff (2002), Hubbard & Brandenburg (2009), Rheinhardt & Brandenburg (2012). Combining machine learning algorithms with anisotropic, local, non-linear models might lead to new insights, and we leave this for future work.

The α^2 dynamo considered in this work is an idealized system but has the advantage that due to maximally helical forcing and periodic boundary conditions, magnetic helicity is both conserved and is gauge invariant (see Blackman 2015). This simplistic setup allows developing analytical theory to explain the origin and saturation of magnetic fields (Pouquet et al. 1976; Blackman & Field 2002). Numerical results seem to be consistent with theory (Brandenburg 2001; Brandenburg et al. 2012). In this work, we considered existing analytical formulations and used machine learning tools to test whether the EMF is linear or non-linear in the mean magnetic fields. Machine learning models can complement analytical theory in looking for reduced descriptions of high dimensional systems (Brunton & Kutz 2019).

5. Conclusions

Our main result can be summarized as follows: because we are in the "small" data regime and we are dealing with ordered data (high signal to noise ratio due to helically forced turbulence), regularized linear regression (LASSO) provides the best fit. For small organized data, sources of randomness such as random train/test splits, bootstrapping, cross validation with random subsets do not help. For this particular dataset, many of the features are strongly correlated with one another complicating the model fits even further. Strong correlations also imply that feature engineering to construct polynomial basis is not particularly useful.

Random forests and MCMC do better with mid-sized to big data. Moreover, these ensemble methods tend to outperform linear and regularized linear regression when the signal to noise ratio is poor (see Table 3). For the data we consider in this paper, it is therefore no surprise that LASSO provides the best fit.

What we intended to demonstrate using the example of α^2 is that sophisticated machine learning algorithms can be used to analyze data from DNS of MHD turbulence thanks to the publicly available machine learning libraries. This provides an exciting opportunity to re-visit some long-standing problems in astrophysical flows where linear regression has dominated modeling efforts. For example, one can look at the full data cube without planar averaging and study whether 3D localized structures play a dominant role in the sustenance of the turbulence. Here, deep learning (Lusch et al. 2018) will be a useful alternative as it is known to outperform classical machine learning methods in image processing, voice recognition, natural language processing. One particular problem of interest is shear-driven dynamos (Tobias & Cattaneo 2013; Nauman & Blackman 2014).

Acknowledgements. We thank A. Brandenburg for several stimulating discussions and comments on an earlier draft of the paper. We also thank the anonymous referee for constructive comments that led to several improvements. The work has been performed under the Project HPC-EUROPA3 (INFRAIA-2016-1-730897), with the support of the EC Research Innovation Action under the H2020 Programme; in particular, the author gratefully acknowledges the support of Dhrubaditya Mitra (NORDITA, Stockholm) and the computer resources and technical support provided by PDC, Stockholm. We used the following python packages: NUMPY (Walt 2011), JUPYTER NOTEBOOK (Kluyver et al. 2016), MATPLOTLIB (Hunter 2007), PANDAS (McKinney 2010), SCIKIT-LEARN (Pedregosa et al. 2011), EMCEE (Foreman-Mackey et al. 2013), MINEPY (Albanese et al. 2012).

References

Albanese, D., Filosi, M., Visintainer, R., et al. 2012, Bioinformatics, 29, 407
Blackman, E. G. 2015, Space Sci. Rev., 188, 59
Blackman, E. G., & Brandenburg, A. 2002, ApJ, 579, 359
Blackman, E. G., & Field, G. B. 2002, Phys. Rep., 89, 265007
Bollen, K. A., & Pearl, J. 2013, in Eight Myths About Causality and Structural Equation Models, ed. S. L. Morgan (Dordrecht, Netherlands: Springer), 301
Brandenburg, A. 2001, ApJ, 550, 824
Brandenburg, A. 2018, J. Plasma Phys., 84, 735840404
Brandenburg, A., & Dobler, W. 2002, Comput. Phys. Commun., 147, 471
Brandenburg, A., & Sokoloff, D. 2002, Geophys. Astrophys. Fluid Dyn., 96, 319
Brandenburg, A., & Subramanian, K. 2005, Phys. Rep., 417, 1
Brandenburg, A., Sokoloff, D., & Subramanian, K. 2012, Space Sci. Rev., 169, 123
Breiman, L. 2001a, Mach. Learn., 45, 5
Breiman, L. 2001b, Statist. Sci., 16, 199
Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. 1984, Classification and Regression Trees (Chapman and Hall/CRC)
Brun, A. S., & Browning, M. K. 2017, Liv. Rev. Sol. Phys., 14, 4
Brunton, S. L., & Kutz, J. N. 2019, Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control (Cambridge University Press)
Charbonneau, P. 2014, ARA&A, 52, 251
Ercolano, B., & Pascucci, I. 2017, R. Soc. Open Sci., 4, 170114
Foreman-Mackey, D., Hogg, D. W., Lang, D., & Goodman, J. 2013, PASP, 125, 306
Géron, A. 2017, Hands-On Machine Learning with Scikit-Learn and TensorFlow (O'Reilly Media)

Grinstead, C., & Snell, J. 1997, Introduction to Probability, 2nd ed. (Providence, RI: American Mathematical Society)
Guido, S., & Müller, A. C. 2017, Introduction to Machine Learning with Python (O'Reilly Media)
Guyon, I., & Elisseeff, A. 2003, J. Mach. Learn. Res., 3, 1157
Hastie, T., Tibshirani, R., & Friedman, J. 2009, The Elements of Statistical Learning (Springer: New York)
Hastie, T., Tibshirani, R., & Wainwright, M. 2015, Statistical Learning with Sparsity (Chapman and Hall/CRC)
Haugen, N. E., Brandenburg, A., & Dobler, W. 2004, Phys. Rev. E, 70, 016308
Hennigh, O. 2017, ArXiv e-prints [arXiv:1705.09036]
Hubbard, A., & Brandenburg, A. 2009, ApJ, 706, 712
Hunter, J. D. 2007, Comput. Sci. Eng., 9, 90
Jepps, S. A. 1975, J. Fluid Mech., 67, 625
Karak, B. B., Rheinhardt, M., Brandenburg, A., Käpylä, P. J., & Käpylä, M. J. 2014, ApJ, 795, 16
Kluyver, T., Ragan-Kelley, B., Pérez, F., et al. 2016, in Positioning and Power in Academic Publishing: Players, Agents and Agendas, eds. F. Loizides, & B. Schmidt (IOS Press), 87
Krause, F., & Rädler, K. H. 1980, Mean-field Magnetohydrodynamics and Dynamo Theory
Krause, F., & Steenbeck, M. 1967, Z. Naturforsch. Teil A, 22, 671
Kutz, J. N. 2013, Data-Driven Modeling & Scientific Computation: Methods for Complex Systems & Big Data (New York: Oxford University Press, Inc.)
Kutz, J. N. 2017, J. Fluid Mech., 814, 1
LeCun, Y., Bengio, Y., & Hinton, G. E. 2015, Nature, 521, 436
Louppe, G. 2014, PhD Thesis, University of Liege, Belgium
Lusch, B., Kutz, J. N., & Brunton, S. L. 2018, Nat. Commun., 9
McKinney, W. 2010, in Proceedings of the 9th Python in Science Conference, eds. S. van der Walt, & J. Millman, 51
Meinshausen, N., & Bühlmann, P. 2010, J. R. Stat. Soc. Ser. B (Stat. Method.), 72, 417
Nauman, F., & Blackman, E. G. 2014, MNRAS, 441, 1855
Navon, I. M. 2009, in Data Assimilation for Numerical Weather Prediction: A Review, eds. S. K. Park, & L. Xu (Berlin, Heidelberg: Springer), 21
Olson, R. S., Cava, W. L., Mustahsan, Z., Varik, A., & Moore, J. H. 2017, Biocomputing 2018 (World Scientific)
Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2011, J. Mach. Learn. Res., 12, 2825
Pouquet, A., Frisch, U., & Leorat, J. 1976, J. Fluid Mech., 77, 321
Raschka, S. 2018, CoRR [arXiv:1811.12808]
Rheinhardt, M., & Brandenburg, A. 2010, A&A, 520, A28
Rheinhardt, M., & Brandenburg, A. 2012, Astron. Nachr., 333, 71
Schrinner, M., Rädler, K.-H., Schmitt, D., Rheinhardt, M., & Christensen, U. R. 2007, Geophys. Astrophys. Fluid Dyn., 101, 81
Shukurov, A., Sokoloff, D., Subramanian, K., & Brandenburg, A. 2006, A&A, 448, L33
Strobl, C., Boulesteix, A.-L., Kneib, T., Augustin, T., & Zeileis, A. 2008, BMC Bioinf., 9, 307
Subramanian, K. 2019, Galaxies, 7, 47
Tibshirani, R. 1996, J. R. Stat. Soc. Ser. B (Method.), 58, 267
Tilgner, A., & Brandenburg, A. 2008, MNRAS, 391, 1477
Tobias, S. M., & Cattaneo, F. 2013, Nature, 497, 463
Vainshtein, S. I., & Cattaneo, F. 1992, ApJ, 393, 165
Walt, S., & v. d., Colbert, S. C., & Varoquaux, G., 2011, Sci. Eng., 13, 22
Wolpert, D. H. 1992, Neural Netw., 5, 241
Wurster, J., & Li, Z.-Y. 2018, Front. Astron. Space Sci., 5, 39