frontiers in Artificial Intelligence

ORIGINAL RESEARCH
published: 28 April 2020
doi: 10.3389/frai.2020.00025

Discovery of Physics From Data: Universal Laws and Discrepancies

Brian M. de Silva[1]*, *David M. Higdon*[2], *Steven L. Brunton*[3] *and J. Nathan Kutz*[1]

[1] Applied Mathematics, University of Washington, Seattle, WA, United States, [2] Department of Statistics, Virginia Polytechnic Institute and State University, Blacksburg, VA, United States, [3] Mechanical Engineering, University of Washington, Seattle, WA, United States

Machine learning (ML) and artificial intelligence (AI) algorithms are now being used to automate the discovery of physics principles and governing equations from measurement data alone. However, positing a universal physical law from data is challenging without simultaneously proposing an accompanying discrepancy model to account for the inevitable mismatch between theory and measurements. By revisiting the classic problem of modeling falling objects of different size and mass, we highlight a number of nuanced issues that must be addressed by modern data-driven methods for automated physics discovery. Specifically, we show that measurement noise and complex secondary physical mechanisms, like unsteady fluid drag forces, can obscure the underlying law of gravitation, leading to an erroneous model. We use the sparse identification of non-linear dynamics (SINDy) method to identify governing equations for real-world measurement data and simulated trajectories. Incorporating into SINDy the assumption that each falling object is governed by a similar physical law is shown to improve the robustness of the learned models, but discrepancies between the predictions and observations persist due to subtleties in drag dynamics. This work highlights the fact that the naive application of ML/AI will generally be insufficient to infer universal physical laws without further modification.

Keywords: dynamical systems, system identification, machine learning, artificial intelligence, sparse regression, discrepancy modeling

OPEN ACCESS

Edited by:
Víctor M. Eguíluz,
Institute of Interdisciplinary Physics and Complex Systems (IFISC), Spain

Reviewed by:
Benson K. Muite,
University of Tartu, Estonia
Roger Guimera,
University of Rovira i Virgili, Spain

Correspondence:
Brian M. de Silva
bdesilva@uw.edu

Specialty section:
This article was submitted to Machine Learning and Artificial Intelligence, a section of the journal Frontiers in Artificial Intelligence

Received: *18 June 2019*
Accepted: *30 March 2020*
Published: *28 April 2020*

Citation:
de Silva BM, Higdon DM, Brunton SL and Kutz JN (2020) Discovery of Physics From Data: Universal Laws and Discrepancies. Front. Artif. Intell. 3:25. doi: 10.3389/frai.2020.00025

1. INTRODUCTION

The ability to derive governing equations and physical principles has been a hallmark feature of scientific discovery and technological progress throughout human history. Even before the scientific revolution, the Ptolemaic doctrine of the *perfect circle* (Peters and Knobel, 1915; Ptolemy, 2014) provided a principled decomposition of planetary motion into a hierarchy of circles, i.e., a bona fide theory for planetary motion. The scientific revolution and the resulting development of calculus provided the mathematical framework and language to precisely describe scientific principles, including gravitation, fluid dynamics, electromagnetism, quantum mechanics, etc. With advances in data science over the past few decades, principled methods are emerging for such scientific discovery from time-series measurements alone. Indeed, across the engineering, physical and biological sciences, significant advances in sensor and measurement technologies have afforded unprecedented new opportunities for scientific exploration. Despite its rapid advancements and wide-spread deployment, *machine learning* (ML) and *artificial intelligence* (AI) algorithms for scientific discovery face significant challenges and limitations, including noisy and corrupt data, latent variables, multiscale physics, and the tendency for overfitting. In this manuscript, we revisit one of the classic problems of physics considered by Galileo and Newton, that of falling objects and gravitation. We demonstrate that a sparse regression framework is well-suited for physics discovery, while highlighting both the need for principled methods to extract parsimonious physics models and the challenges associated with the naive application of ML/AI techniques. Even this simplest of physical examples demonstrates critical principles that must be considered in order to make data-driven discovery viable across the sciences.

Measurements have long provided the basis for the discovery of governing equations. Through empirical observations of planetary motion, the Ptolemaic theory of motion was developed (Peters and Knobel, 1915; Ptolemy, 2014). This was followed by Kepler's laws of planetary motion and the elliptical courses of planets in a heliocentric coordinate system (Kepler, 2015). By hand calculation, he was able to regress Brahe's state-of-the-art data on planetary motion to the minimally parameterized elliptical orbits which described planetary orbits with a terseness the Ptolemaic system had never managed to achieve. Such models led to the development of Newton's $\mathbf{F} = m\mathbf{a}$ (Newton, 1999), which provided a universal, generalizable, interpretable, and succinct description of physical dynamics. Parsimonious models are critical in the philosophy of Occam's razor: the simplest set of explanatory variables is often the best (Blumer et al., 1987; Domingos, 1999; Bongard and Lipson, 2007; Schmidt and Lipson, 2009). It is through such models that many technological and scientific advancements have been made or envisioned.

What is largely unacknowledged in the scientific discovery process is the intuitive leap required to formulate physics principles and governing equations. Consider the example of falling objects. According to physics folklore, Galileo discovered, through experimentation, that objects fall with the same constant acceleration, thus disproving Aristotle's theory of gravity, which stated that objects fall at different speeds depending on their mass. The leaning tower of Pisa is often the setting for this famous stunt, although there is little evidence such an experiment actually took place (Cooper, 1936; Adler and Coulter, 1978; Segre, 1980). Indeed, many historians consider it to have been a thought experiment rather than an actual physical test. Many of us have been to the top of the leaning tower and have longed to drop a bowling ball from the top, perhaps along with a golf ball and soccer ball, in order to replicate this experiment. If we were to perform such a test, here is what we would likely find: Aristotle was correct. Balls of different masses and sizes *do* reach the ground at different times. As we will show from our own data on falling objects, (noisy) experimental measurements may be insufficient for discovering a constant gravitational acceleration, especially when the objects experience Reynolds numbers varying by orders of magnitudes over the course of their trajectories. But what is beyond dispute is that Galileo did indeed *posit* the idea of a fixed acceleration, a conclusion that would have been exceptionally difficult to come to from such measurement data alone. Gravitation is only one example of the intuitive leap required for a paradigm shifting physics discovery. Maxwell's equations (Maxwell, 1873) have a similar story arc revolving around Coulomb's inverse square law. Maxwell cited Coulomb's torsion balance experiment as establishing the inverse square law while dismissing it only a few pages later as an approximation (Bartlett et al., 1970; Falconer, 2017). Maxwell concluded that Faraday's observation that an electrified body, touched to the inside of a conducting vessel, transfers all its electricity to the outside surface as much more direct proof of the square law. In the end, both would have been approximations, with Maxwell taking the intuitive leap that exactly a power of negative two was needed when formulating Maxwell's equations. Such examples abound across the sciences, where intuitive leaps are made and seminal theories result.

One challenge facing ML and AI methods is their inability to take such leaps. At their core, many ML and AI algorithms involve regressions based on data, and are statistical in nature (Breiman, 2001; Bishop, 2006; Wu et al., 2008; Murphy, 2012). Thus by construction, a model based on measurement data would not produce an exact inverse *square* law, but rather a slightly different estimate of the exponent. In the case of falling objects, ML and AI would yield an Aristotelian theory of gravitation, whereby the data would suggest that objects fall at a speed related to their mass. Of course, even Galileo intuitively understood that air resistance plays a significant role in the physics of falling objects, which is likely the reason he conducted controlled experiments on inclined ramps. Although we understand that air resistance, which is governed by latent fluid dynamic variables, explains the discrepancy between the data and a constant gravity model, our algorithms do not. Without modeling these small disparities (e.g., due to friction, heat dissipation, air resistance, etc.), it is almost impossible to uncover universal laws, such as gravitation. Differences between theory and data have played a foundational role in physics, with general relativity arising from inconsistencies between gravitational theory and observations, and quantum mechanics arising from our inability to explain the photoelectric effect with Maxwell's equations.

Our goal in this manuscript is to highlight the many subtle and nuanced concerns related to data-driven discovery using modern ML and AI methods. Specifically, we highlight these issues on the most elementary of problems: modeling the motion of falling objects. Given our ground-truth knowledge of the physics, this example provides a convenient testbed for different physics discovery techniques. It is important that one clearly understands the potential pitfalls in such methods before applying them to more sophisticated problems which may arise in fields like biology, neuroscience, and climate modeling. Our physics discovery method is rooted in the *sparse identification for non-linear dynamics* (SINDy) algorithm, which has been shown to extract parsimonious governing equations in a broad range of physical sciences (Brunton et al., 2016). SINDy has been widely applied to identify models for fluid flows (Loiseau and Brunton, 2018; Loiseau et al., 2018), optical systems (Sorokina et al., 2016), chemical reaction dynamics (Hoffmann et al., 2019), convection in a plasma (Dam et al., 2017), structural modeling (Lai and Nagarajaiah, 2019), and for model predictive control (Kaiser et al., 2018). There are also a number of theoretical extensions to the SINDy framework, including for identifying partial differential equations (Rudy et al., 2017; Schaeffer, 2017), and models with rational function non-linearities (Mangan et al., 2016). It can also incorporate partially known physics and constraints (Loiseau and Brunton, 2018). The algorithm can be reformulated to include integral terms for noisy data (Schaeffer and McCalla, 2017) or handle incomplete or limited data (Tran and Ward, 2016; Schaeffer et al., 2018). In this manuscript we show that *group sparsity* (Rudy et al., 2019) may be used to enforce that the same model terms explain *all* of the observed trajectories, which is essential in identifying the correct model terms without overfitting.

SINDy is by no means the only attempt that has been made at using machine learning to infer physical models from data. Gaussian processes have been employed to learn conservation laws described by parametric linear equations (Raissi et al., 2017a). Symbolic regression has been successfully applied to the problem of inferring dynamics from data (Bongard and Lipson, 2007; Schmidt and Lipson, 2009). Another closely related set of approaches are process-based models (Bridewell et al., 2008; Tanevski et al., 2016, 2017) which, similarly to SINDy, allow one to specify a library of relationships or functions between variables based on domain knowledge and produce an interpretable set of governing equations. The principal difference between process-based models and SINDy is that SINDy employs sparse regression techniques to perform function selection which allows a larger class of library functions to be considered than is tractable for process-based models. Deep learning methods have been proposed for accomplishing a variety of related tasks, such as predicting physical dynamics directly (Mrowca et al., 2018), building neural networks that respect given physical laws (Raissi et al., 2017b), discovering parameters in non-linear partial differential equations with limited measurement data (Raissi et al., 2017c), and simultaneously approximating the solution and non-linear dynamics of non-linear partial differential equations (Raissi, 2018). Graph neural networks (Battaglia et al., 2018), a specialized class of neural networks that operate on graphs, have been shown to be effective at learning basic physics simulators from measurement data (Battaglia et al., 2016; Chang et al., 2016) and directly from videos (Watters et al., 2017). It should be noted that the aforementioned neural network approaches either require detailed prior knowledge of the form of the underlying differential equations or fail to yield simple sets of interpretable governing equations.

2. MATERIALS AND METHODS

2.1. Fluid Forces on a Sphere: A Brief History

It must have been immediately clear to Galileo and Newton that committing to a gravitational constant created an inconsistency with experimental data. Specifically, one had to explain why objects of different sizes and shapes fall at different speeds (e.g., a feather vs. a cannon ball). Wind resistance was an immediate candidate to explain the *discrepancy* between a universal gravitational constant and measurement data. The fact that Galileo performed experiments where he rolled balls down inclines seems to suggest that he was keenly aware of the need to isolate and disambiguate the effects of gravitational forces from fluid drag forces. Discrepancies between the Newtonian theory of gravitation and observational data of Mercury's orbit led to Einstein's development of general relativity. Similarly, the photoelectric effect was a discrepancy in Maxwell's equations which led to the development of quantum mechanics.

Discrepancy modeling is therefore a critical aspect of building and discovering physical models. Consider the motion of falling spheres as a prototypical example. In addition to the force of gravity, a falling sphere encounters a fluid drag force as it passes through the air. A simple model of the drag force F_D is given by:

$$F_D = \frac{1}{2}\rho v^2 A C_D, \tag{1}$$

where ρ is the fluid density, v is the velocity of the sphere with respect to the fluid, $A = \pi D^2/4$ is the cross-sectional area of the sphere, D is the diameter of the sphere, and C_D is the dimensionless drag coefficient. As the sphere accelerates through the fluid, its velocity increases, exciting various unsteady aerodynamic effects, such as laminar boundary layer separation, vortex shedding, and eventually a turbulent boundary layer and wake (Moller, 1938; Magarvey and MacLatchy, 1965; Achenbach, 1972, 1974; Calvert, 1972; Smits and Ogg, 2004). Thus, the drag coefficient is a function of the sphere's velocity, and this coefficient generally decreases for increasing velocity. **Figure 1** shows the drag coefficient C_D for a sphere as a function of the Reynolds number $Re = \rho v D/\mu$, where μ is the dynamic viscosity of the fluid; for a constant diameter and viscosity, the Reynolds number is directly proportional to the velocity. Note that the drag coefficient of a smooth sphere will differ from that of a rough sphere. The flow over a rough sphere will become turbulent at lower velocities, causing less flow separation and a more streamlined, lower-drag wake; this explains why golf balls are dimpled, so that they will travel farther (Smits and Ogg, 2004). Thus, (1) states that drag is related to the square of the velocity, although C_D has a weak dependence on velocity. When Re is small, C_D is proportional to $1/v$, resulting in a drag force that is linear in v. For larger values of Re, C_D is approximately constant (away from the steep drop), leading to a quadratic drag force. Eventually, the drag force will balance the force of gravity, resulting in the sphere reaching its *terminal velocity*. In addition, as the fluid wake becomes unsteady, the drag force will also vary in time, although these variations are typically fast and may be time-averaged. Finally, objects accelerating in a fluid will also accelerate the fluid out of the way, resulting in an effective mass that includes the mass of the body and an *added mass* of accelerated fluid (Newman, 1977); however, this added mass force will typically be quite small in air.

In addition to the theoretical study of fluid forces on an idealized sphere, there is a rich history of scientific inquiry into the aerodynamics of sports balls (Mehta, 1985, 2008; Smits and Ogg, 2004; Goff, 2013). Apart from gravity and drag, a ball's trajectory can be influenced by the spin of the ball via the Magnus force or lift force which acts in a direction orthogonal to the drag. Other factors that can affect the forces experienced by a

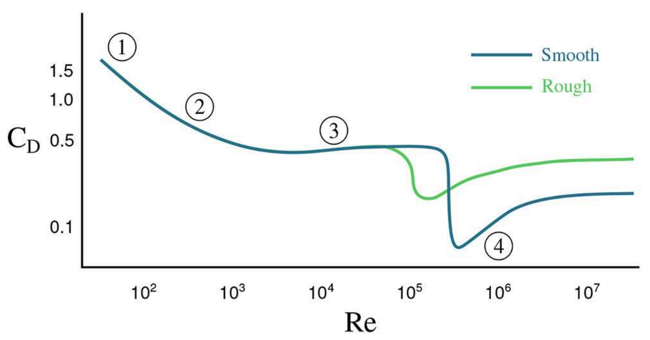

FIGURE 1 | The drag coefficient for a sphere as a function of Reynolds number, *Re*. The dark curve shows the coefficient for a sphere with a smooth surface and the light curve a sphere with a rough surface. The numbers highlight different flow regimes. (1) attached flow and steady separated flow; (2) separated unsteady flow, with laminar flow boundary layer upstream of separation, producing a Kármán vortex street; (3) separated unsteady flow with a chaotic turbulent wake downstream and a laminar boundary layer upstream; (4) post-critical separated flow with turbulent boundary layer.

falling ball include air temperature, wind, elevation, and ball surface shape.

2.2. Data Set

The data considered in this manuscript are height measurements of balls falling through air. These measurements originate from two sources: physical experiments and simulations. Such experiments are popular in undergraduate physics classes where they are used to explore linear vs. quadratic drag (Owen and Ryu, 2005; Kaewsutthi and Wattanakasiwich, 2011; Christensen et al., 2014; Cross and Lindsey, 2014) and scaling laws Sznitman et al. (2017). In June 2013 a collection of balls, pictured in **Figure 2**, were dropped, *twice each*, from the Alex Fraser Bridge in Vancouver, BC from a height of about 35 meters above the landing site. In total 11 balls were dropped: a golf ball, a baseball, two whiffle balls with elongated holes, two whiffle balls with circular holes, two basketballs, a bowling ball, and a volleyball (not pictured). More information about the balls is given in **Table 1**. The air temperature at the time of the drops was 65°F (18°C). A hand held iPad was used to record video of the drops at a rate of 15 frames per second. The height of the falling objects was then estimated by tracking the balls in the resulting videos. **Figure 3** visualizes the second set of ball drops. As one might expect, the whiffle balls all reach the ground later than the other balls. This is to be expected since the openings in their faces increase the drag they experience. Even so, all the balls reach the ground within a second of each other. We also plot the simulated trajectories of two spheres falling with constant linear (in v) drag and the trajectory predicted by constant acceleration. Note that, based on the log-log plot of displacement, none of the balls appears to have reached terminal velocity by the time they hit the ground. This may increase the difficulty of accurately inferring the balls' governing equations. Given only measurements from one regime of falling ball dynamics, it may prove difficult to infer models that generalize to other regimes.

Drawing inspiration from Aristotle, one might form the hypothesis that the amount of time taken by spheres to reach the ground should be a function of the *density* of the spheres. Density takes into account both information about the mass of an object and its volume, which might be thought to affect the air resistance it encounters. We plot the landing time of each ball as a function of its density for both drops in **Figure 4**. To be more precise, because some balls were dropped from slightly different heights, we measure the amount of time it takes each ball to travel a fixed distance after being dropped, not the amount of time it takes the ball to reach the ground. There is a general trend across the tests for the denser balls to travel faster. However, the basketballs defy this trend and complete their journeys about as quickly as the densest ball. This shows there must be more factors at play than just density. There is also variability in the land time of the balls across drops. While most of the balls have very consistent fall times across drops, the blue basketball, golf ball, and orange whiffle ball reach the finish line faster in the first trial than the second one. These differences could be due to a variety of factors, including the balls being released with different initial velocities, or errors in measuring the balls' heights.

There are multiple known sources of error in the measurement data. The relatively low resolution of the videos means that the inferred ball heights are only approximate. In the **Supplementary Material** we attempt to infer the level of noise introduced by our use of heights derived from imperfect video data. Furthermore, the camera was held by a person, not mounted on a tripod, leading to shaky footage. The true bridge height is uncertain because it was measured with a laser range finder claiming to be accurate to within 0.5 m. Because the experiments were executed outside, it is possible for any given drop to have been affected by wind. Detecting exactly when each ball was dropped, at what velocity it was dropped, and when it hit the ground using only videos is certain to introduce further error. Finally, treating these balls as perfect spheres is an approximation whose accuracy depends on the nature of the balls. This idealization seems least appropriate for the whiffle balls, which are sure to exhibit much more complicated aerodynamic effects than, say, the baseball. The bowling ball was excluded from consideration because of corrupted measurements from its first drop.

The situation we strive to mimic with this experiment is one in which the researcher is in a position of ignorance about the system being studied. In order to design an experiment which eliminates the effects of confounding factors, such as air resistance one must already have an appreciation for which factors are worth controlling; one leverages prior knowledge as Galileo did when he employed ramps in his study of falling objects to mitigate the effect of air resistance. In the early stages of investigation of a physical phenomenon, one must often perform poorly-controlled experiments to help identify these factors. We view the ball drop trials as this type of experiment.

In addition to the measurement data just described, we construct a synthetic data set by simulating falling objects with masses of 1 kg and different (linear) drag coefficients. In particular, for each digital ball, we simulate two drops of the same length as the real data and collect height measurements at

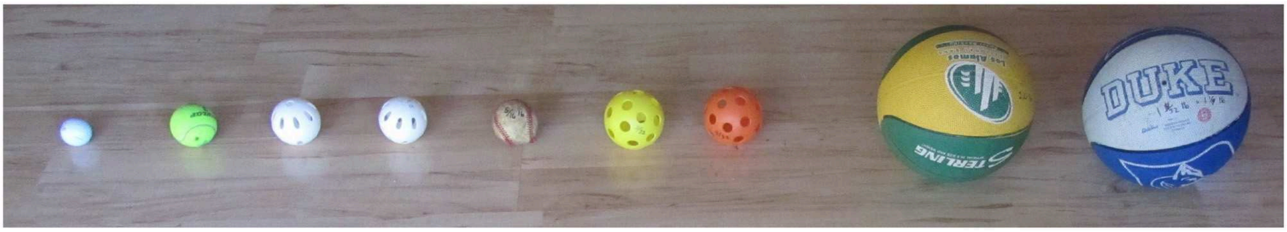

FIGURE 2 | The balls that were dropped from the bridge, with the volleyball omitted. From left to right: Golf Ball, Tennis Ball, Whiffle Ball 1, Whiffle Ball 2, Baseball, Yellow Whiffle Ball, Orange Whiffle Ball, Green Basketball, and Blue Basketball. The two colored whiffle balls have circular openings and are structurally identical. The two white whiffle balls have elongated slits and are also identical.

TABLE 1 | Physical measurements, maximum velocities across the two drops, and maximum Reynolds numbers for the dropped balls.

Ball	Radius (m)	Mass (kg)	Density (kg/m)	Max vel. (m/s)	Max Re
Golf ball	0.021963	0.045359	1022.066427	26.63	1.75×10^5
Baseball	0.035412	0.141747	762.037525	26.61	2.83×10^5
Tennis ball	0.033025	0.056699	375.813253	21.95	2.18×10^5
Volleyball	0.105*	NA	NA	22.09	6.96×10^5
Blue basketball	0.119366	0.510291	71.628378	24.80	8.88×10^5
Green basketball	0.116581	0.453592	68.342914	25.06	8.77×10^5
Whiffle ball 1	0.036287	0.028349	141.641937	16.91	1.84×10^5
Whiffle ball 2	0.036287	0.028349	141.641937	16.35	1.78×10^5
Yellow whiffle ball	0.046155	0.042524	103.250857	15.30	2.12×10^5
Orange whiffle ball	0.046155	0.042524	103.250857	15.77	2.18×10^5

We do not have measurement data for the volleyball, but obtained an estimate for its radius based on other volleyballs in order to approximate its maximum Reynolds number.

a rate of 15 measurements per second. The balls fall according to the equation $\ddot{x}(t) = -9.8 + D\dot{x}(t)$, with each ball having its own *constant* drag coefficient, $D < 0$. We simulate five balls in total, with respective drag coefficients −0.1, −0.3, −0.3, −0.5, and −0.7. These coefficients are all within the plausible range suggested by the simulated trajectories shown in **Figure 3**. Each object is "dropped" with an initial velocity of 0. Varying amounts of Gaussian noise are added to the height data so that we may better explore the noise tolerance of the proposed model discovery approaches:

$$\tilde{x}_i = x_i + \eta\epsilon_i.$$

where $\eta \geq 0$ and $\epsilon_i \sim N(0, 1)$; that is to say ϵ_i is normally distributed with unit variance.

2.3. Methods

In this section we describe the model discovery methods we employ to infer governing equations from noisy data. We first give the mathematical background necessary for learning dynamics via sparse regression and provide a brief overview of the SINDy method in section 2.3.1. In section 2.3.2 we propose a group sparsity regularization strategy for improving the robustness and generalizability of SINDy. We briefly discuss the setup of the model discovery problem we are attempting to solve in section 2.3.3. Finally, we discuss numerical differentiation, a subroutine critical to effective model discovery, in section 2.3.4.

2.3.1. Sparse Identification of Non-linear Dynamical Systems

Consider the non-linear dynamical system for the state vector $\mathbf{x}(t) = [x_1(t), x_2(t), \dots, x_n(t)]^\top \in \mathbb{R}^n$ defined by

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t)).$$

Given a set of noisy measurements of $\mathbf{x}(t)$, the sparse identification of non-linear dynamics (SINDy) method, introduced in Brunton et al. (2016), seeks to identify $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$. In this section we give an overview of the steps involved in the SINDy method and the assumptions upon which it relies. Throughout this manuscript we refer to this algorithm as the *unregularized SINDy* method, not because it involves no regularization, but because its regularization is not as closely tailored to the problem at hand as the method proposed in section 2.3.2.

For many dynamical systems of interest, the function specifying the dynamics, $\mathbf{f}$, consists of only a few terms. That is to say, when represented in the appropriate basis, there is a sense in which it is sparse. The key idea behind the SINDy method is that if one supplies a rich enough set of candidate functions for representing $\mathbf{f}$, then the correct terms can be identified using sparse regression techniques. The explicit steps are as follows. First we collect a set of (possibly noisy) measurements of the state $\mathbf{x}(t)$ and its derivative $\dot{\mathbf{x}}(t)$ at a sequence of points in time, $t_1, t_2, \dots, t_m$. These measurements are concatenated into two matrices, the columns of which correspond to different

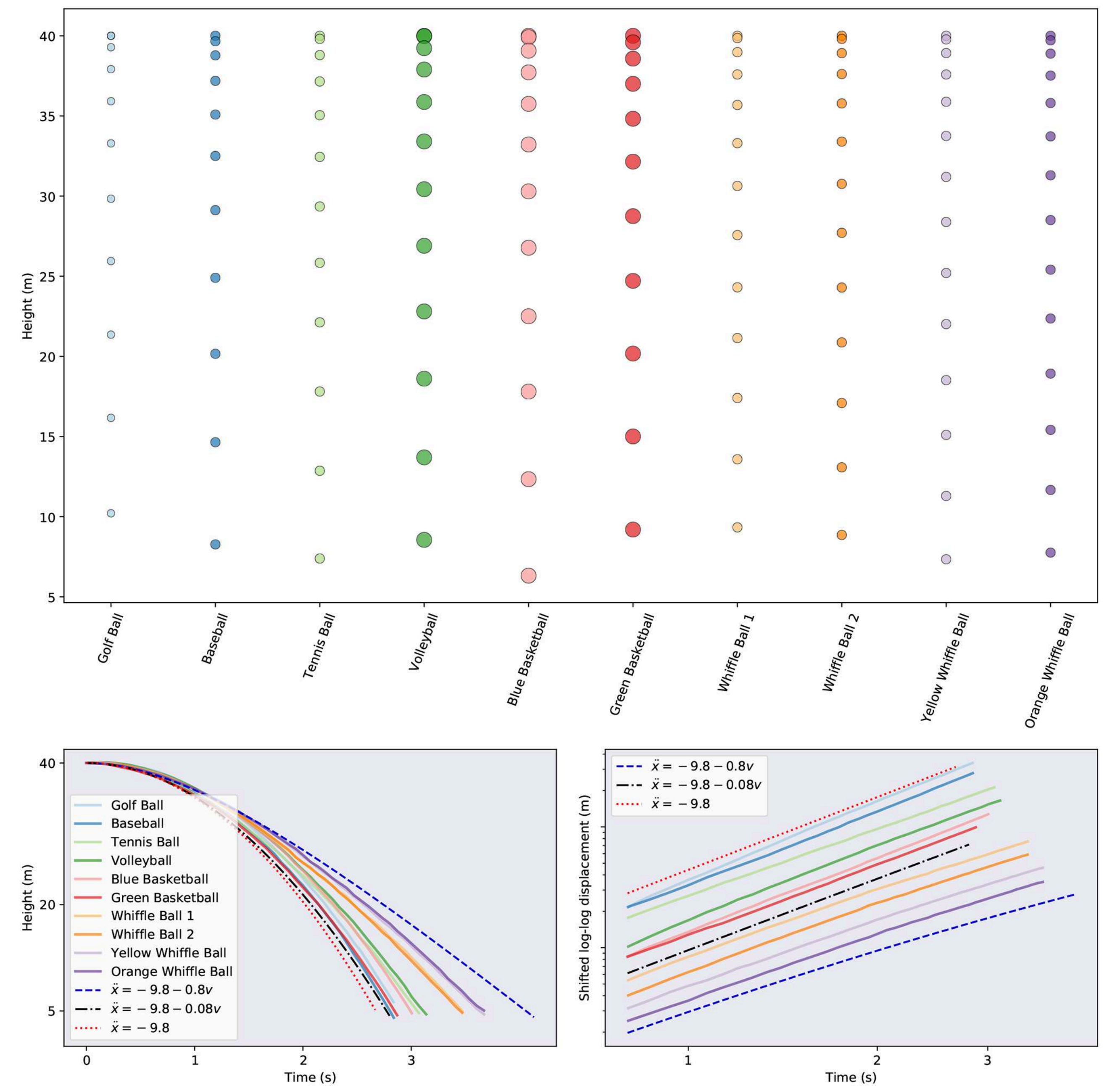

FIGURE 3 | Visualizations of the ball trajectories for the second drop. (Top) Subsampled raw drop data for each ball. (Bottom left) Height for each ball as a function of time. We also include the simulated trajectories of idealized balls with differing levels of drag (black and blue) and a ball with constant acceleration (red). (Bottom right) A log-log plot of the displacement of each ball from its original position atop the bridge. Note that we have shifted the curves vertically and zoomed in on the later segments of the time series to enable easier comparison. In this plot a ball falling at a constant rate (zero acceleration) will have a trajectory represented by a line with slope one. A ball falling with constant acceleration will have a trajectory represented by a line with slope two. A ball with drag will have a trajectory which begins with slope two and asymptotically approaches a line with slope one.

state variables and the rows of which correspond to points in time.

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}(t_1)^\top \\ \mathbf{x}(t_2)^\top \\ \vdots \\ \mathbf{x}(t_m)^\top \end{bmatrix} = \begin{bmatrix} x_1(t_1) & x_2(t_1) & \dots & x_n(t_1) \\ x_1(t_2) & x_2(t_2) & \dots & x_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(t_m) & x_2(t_m) & \dots & x_n(t_m) \end{bmatrix},$$

$$\dot{\mathbf{X}} = \begin{bmatrix} \dot{\mathbf{x}}(t_1)^\top \\ \dot{\mathbf{x}}(t_2)^\top \\ \vdots \\ \dot{\mathbf{x}}(t_m)^\top \end{bmatrix} = \begin{bmatrix} \dot{x}_1(t_1) & \dot{x}_2(t_1) & \dots & \dot{x}_n(t_1) \\ \dot{x}_1(t_2) & \dot{x}_2(t_2) & \dots & \dot{x}_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \dot{x}_1(t_m) & \dot{x}_2(t_m) & \dots & \dot{x}_n(t_m) \end{bmatrix}.$$

Next we specify a set of candidate functions, $\{\phi_i(\mathbf{x}) : i = 1, 2, \dots, p\}$, with which to represent $\mathbf{f}$. Examples of candidate

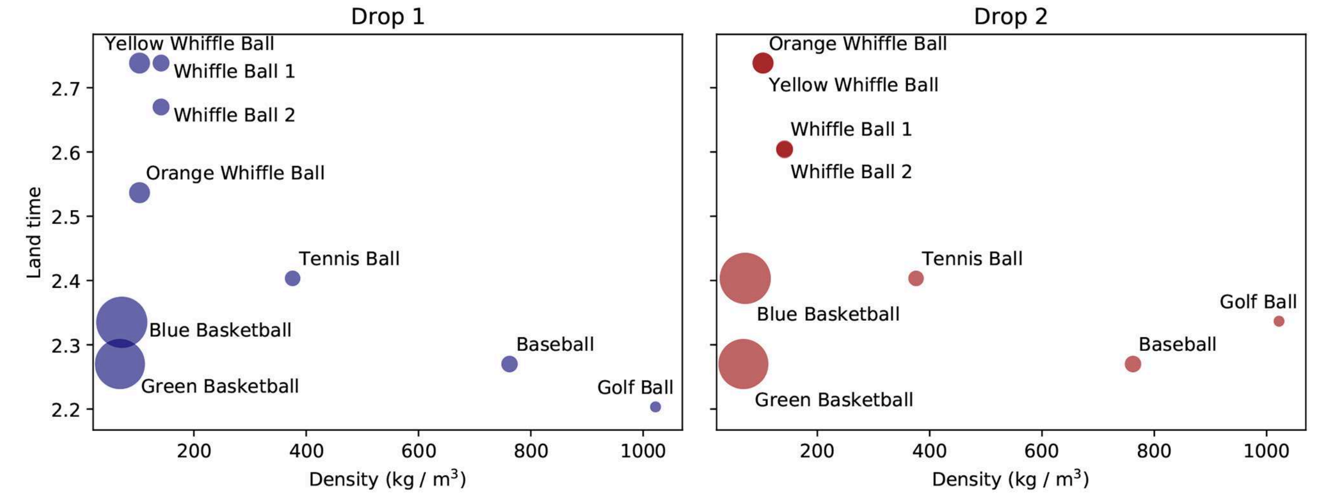

FIGURE 4 | The amount of time taken by each ball to travel a fixed distance as a function of ball density.

functions include monomials up to some finite degree, trigonometric functions, and rational functions. In practice the selection of these functions can be informed by the practitioner's prior knowledge about the system being measured. The candidate functions are evaluated on $\mathbf{X}$ to construct a library matrix

$$\Phi(\mathbf{X}) = \begin{bmatrix} | & | & & | \\ \phi_1(\mathbf{X}) & \phi_2(\mathbf{X}) & \dots & \phi_p(\mathbf{X}) \\ | & | & & | \end{bmatrix}.$$

Note that each column of $\Phi(\mathbf{X})$ corresponds to a single candidate function. Here we have overloaded notation and interpret $\phi(\mathbf{X})$ as the column vector obtained by applying ϕ_i to each row of $\mathbf{X}$. It is assumed that each component of $\mathbf{f}$ can be represented as a *sparse* linear combination of such functions. This allows us to pose a regression problem to be solved for the coefficients used in these linear combinations:

$$\dot{\mathbf{X}} = \Phi(\mathbf{X})\Xi. \tag{2}$$

We adopt MATLAB-style notation and use $\Xi_{(:,j)}$ to denote the j-th column of Ξ. The coefficients specifying the dynamical system obeyed by $\mathbf{x}_j$ are stored in $\Xi_{(:,j)}$:

$$\dot{\mathbf{x}}_j = \mathbf{f}_j(\mathbf{x}) = \Phi\left(\mathbf{x}^\top\right)\Xi_{(:,j)},$$

where $\Phi\left(\mathbf{x}^\top\right)$ is to be interpreted as a (row) vector of symbolic functions of components of $\mathbf{x}$. The full system of differential equations is then given by

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) = \Xi^\top\left(\Phi\left(\mathbf{x}^\top\right)\right)^\top.$$

For concreteness we supply the following example. With the candidate functions $\left\{1, x_1, x_2, x_1x_2, x_1^2, x_2^2\right\}$ the Lotka-Volterra equations

$$\begin{cases} \dot{x}_1 = & \alpha x_1 - \beta x_1 x_2, \\ \dot{x}_2 = & \delta x_1 x_2 - \gamma x_2 \end{cases}$$

can be expressed as

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \Xi^\top\left(\Phi\left(\mathbf{x}^\top\right)\right)^\top = \begin{bmatrix} 0 & \alpha & 0 & -\beta & 0 & 0 \\ 0 & 0 & -\gamma & \delta & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_1x_2 \\ x_1^2 \\ x_2^2 \end{bmatrix}$$

Were we to obtain pristine samples of $\mathbf{x}(t)$ and $\dot{\mathbf{x}}(t)$ we could solve (2) exactly for Ξ. Furthermore, assuming we chose linearly independent candidate functions and avoided collecting redundant measurements, Ξ would be unique and would exhibit the correct sparsity pattern. In practice, however, measurements are contaminated by noise and we actually observe a perturbed version of $\mathbf{x}(t)$.

In many cases $\dot{\mathbf{x}}(t)$ is not observed directly and must instead be approximated from $\mathbf{x}(t)$, establishing another source of error. The previously exact Equation (2), to be solved for Ξ is supplanted by the approximation problem

$$\dot{\mathbf{X}} \approx \Phi(\mathbf{X})\Xi.$$

To find Ξ we solve the more concrete optimization problem

$$\min_{\Xi} \frac{1}{2}\left\|\dot{\mathbf{X}} - \Phi(\mathbf{X})\Xi\right\|_F^2 + \Omega(\Xi), \tag{3}$$

where $\Omega(\cdot)$ is a regularization term chosen to promote sparse solutions and $\|\cdot\|_F$ is the Frobenius norm. Note that because any given column of Ξ encodes a differential equation for a single component of $\mathbf{x}$, each column generates a problem that is decoupled from the problems associated with the other columns. Thus, solving (3) consists of solving n separate regularized least squares problems. Row i of Ξ contains the coefficients of library function ϕ_i for each governing equation.

The most direct way to enforce sparsity is to choose Ω to be the ℓ_0 penalty, defined as $\|\mathbf{M}\|_0 = \sum_{i,j}|\text{sign}(M_{ij})|$. This penalty simply counts the number of non-zero entries in a matrix or vector. However, using the ℓ_0 penalty makes (3) difficult to optimize because $\|\cdot\|_0$ is non-smooth and non-convex. Another common choice is the ℓ_1 penalty defined by $\|\mathbf{M}\|_1 = \sum_{i,j}|M_{ij}|$. This function is the convex relaxation of the ℓ_0 penalty. The LASSO, proposed in Tibshirani (1996), with coordinate descent is typically employed to solve (3) with $\Omega(\cdot) = \|\cdot\|_1$, but this method can become computationally expensive for large data sets and often leads to incorrect sparsity patterns (Su et al., 2017). Hence we solve (3) using the sequential thresholded least-squares algorithm proposed in Brunton et al. (2016), and studied in further detail in Zheng et al. (2018). In essence, the algorithm alternates between (a) successively solving the *unregularized* least-squares problem for each column of Ξ and (b) removing candidate functions from consideration whose corresponding components in Ξ are below some threshold. This threshold or sparsity parameter, is straightforward to interpret: no governing equations are allowed to have any terms with coefficients of magnitude smaller than the threshold. Crucially, it should be noted that just because a candidate function is discarded for one column of Ξ (i.e., for one component's governing equation) does not mean it is removed from contention for the other columns. A simple Python implementation of sequentially thresholded least-squares is provided in the **Supplementary Material**.

We note that if we simulate falling objects with constant acceleration, $\ddot{x}(t) = -9.8$, or linear drag, $\ddot{x}(t) = -9.8 + D\dot{x}(t)$, and add *no noise*, then there is almost perfect agreement between the true governing equations and the models learned by SINDy. The **Supplementary Material** contains a more thorough discussion of such numerical experiments and another example application of SINDy.

SINDy has a number of well-known limitations. The biggest of these is the effect of noise on the learned equations. If one does not have direct measurements of derivatives of state variables, then these derivatives must be computed numerically. Any noise that is present in the measurement data is amplified when it is numerically differentiated, leading to noise in both $\dot{\mathbf{X}}$ and $\Phi(\mathbf{X})$ in (3). In its original formulation, SINDy often exhibits erratic performance in the face of such noise, but extensions have been developed which handle noise more gracefully (Tran and Ward, 2016; Schaeffer and McCalla, 2017). We discuss numerical differentiation further in section 2.3.4. As with other methods, each degree of freedom supplied to the practitioner presents a potential source of difficulty. To use SINDy one must select a set of candidate functions, a sparse regularization function, and a parameter weighing the relative importance of the sparseness of the solution against accuracy. An improper choice of any one of these can lead to poor performance. The set of possible candidate functions is infinite, but SINDy requires one to specify a finite number of them. If one has any prior knowledge of the dynamics of the system being modeled, it can be leveraged here. If not, it is typically recommended to choose a class of functions general enough to encapsulate a wide variety of behaviors (e.g., polynomials or trigonometric functions). In theory, sparse regression techniques should allow one to specify a sizable library of functions, selecting only the relevant ones. However, in practice, the underlying regression problem becomes increasingly ill-conditioned as more functions are added. If one wishes to explore an especially large space of possible library functions it may be better to use other approaches, such as symbolic regression with genetic algorithms (Bongard and Lipson, 2007; Schmidt and Lipson, 2009). A full discussion of how to pick a sparsity-promoting regularizer is beyond the scope of this work. We do note that there have been recent efforts to explore different methods for obtaining sparse solutions when using SINDy (Champion et al., 2019). An appropriate value for the sparsity hyperparameter can be obtained using cross-validation. We note that the need to perform hyperparameter tuning is by no means unique to SINDy. Virtually all machine learning methods require some amount of hyperparameter tuning. There are two natural options for target metrics during cross-validation. The derivatives directly predicted by the linear model can be compared against the measured (or numerically computed) derivatives. Alternatively, the model can be fed into a numerical integrator along with initial conditions to obtain predicted future values for the state variables. These forecasts can then be judged against the measured values. To achieve a balance between model sparsity and accuracy, information theoretic criteria, such as the Akaike information criteria (AIC) or Bayes information criteria (BIC) can be applied (Mangan et al., 2017).

2.3.2. Group Sparsity Regularization

The standard, unregularized SINDy approach attempts to learn the dynamics governing each state variable independently. It does not take into account prior information one may possess regarding relationships between state variables. Intuitively speaking, the balls in our data set (whiffle balls, perhaps, excluded) are similar enough objects that the equations governing their trajectories should include similar terms. In this subsection we propose a group sparsity method which can be interpreted as enforcing this hypothesis when seeking predictive models for the balls.

We draw inspiration for our approach from the group LASSO of Yuan and Lin (2006), which extends the LASSO. The classic LASSO method solves the ℓ_1 regularization problem

$$\beta = \arg\min_{\beta} \frac{1}{2}\|\mathbf{X}\beta - \mathbf{Y}\|_2^2 + \lambda\|\beta\|_1. \tag{4}$$

which penalizes the magnitude of each component of β *individually*. The group LASSO approach modifies (4) by bundling sets of related entries of β together when computing the penalty term. Let the entries of β be partitioned into G disjoint blocks $\{\beta_1, \beta_2, \ldots, \beta_G\}$, which can be treated as vectors. The group LASSO then solves the following optimization problem

$$\beta = \arg\min_{\beta} \frac{1}{2}\|\mathbf{X}\beta - \mathbf{Y}\|_2^2 + \lambda\sum_{i=1}^{G}\|\beta_i\|_2. \tag{5}$$

In the case that the groups each consist of exactly one entry of β, (5) reduces to (4). When blocks contain multiple

Algorithm 1: A group sparsity algorithm for the sequential thresholded least squares method.

```
Data: $\dot{\mathbf{X}} \in \mathbb{R}^{m\times d}$, $\Phi(\mathbf{X}) \in \mathbb{R}^{m\times p}$, and $\delta > 0$
Result: coefficient matrix $\Xi \in \mathbb{R}^{p\times d}$
while not converged do
    // Solve a least squares problem for
       each state variable
    for $j \leftarrow 1$ to $d$ do
        $\Xi_{(:,j)} \leftarrow \arg\min_\xi \frac{1}{2}\left\|\dot{\mathbf{X}} - \Phi(\mathbf{X})\xi\right\|_2^2$;
    end
    // Remove library functions with low
       salience
    for $i \leftarrow 1$ to $p$ do
        if $R(\Xi_{(i,:)}) < \delta$ then
            Delete $\Xi_{(i,:)}$ and $\Phi(\mathbf{X})_{(:,i)}$;
        end
    end
end
Replace deleted rows of $\Xi$ and deleted columns of $\Phi(\mathbf{X})$
with 0's;
```

entries, the group LASSO penalty encourages them to be retained or eliminated as a group. Furthermore, it drives sets of unimportant variables to truly vanish, unlike the ℓ_2 regularization function which merely assigns small but non-zero values to insignificant variables.

We apply similar ideas in our *group sparsity* method for the SINDy framework and force the models learned for each ball to select the same library functions. Recall that the model variables are contained in Ξ. To enforce the condition that each governing equation should involve the same terms, we identify *rows* of Ξ as sets of variables to be grouped together. Borrowing MATLAB notation again, we let $\Xi_{(i,:)}$ denote row i of Ξ. To perform sequential thresholded least squares with the group sparsity constraint we repeatedly apply the following steps until convergence: (a) solve the least-squares problem (3) *without* a regularization term for each column of Ξ (i.e., for each ball), (b) prune the library, $\Phi(\mathbf{X})$, of functions which have low relevance across most or all of the balls. This procedure is summarized in Algorithm 1.

Here R is a function measuring the importance of a row of coefficients. Possible choices for R include the ℓ_1 or ℓ_2 norm of the input, the mean or median of the absolute values of the entries of the input, or another statistical property of the input entries, such as the lower 25% quantile. In this work we use the ℓ_1 norm. Convergence is attained when no rows of Ξ are removed. Note that while all the models are constrained to be generated by the same library functions, the *coefficients* in front of each can differ from one model to the next. The hyperparameter δ controls the sparsity of Ξ, though not as directly as the sparsity parameter for SINDy. Increasing it will result in models with fewer terms and decreasing it will have the opposite effect. Since we use the ℓ_1 norm and there are 10 balls in our primary data set, rows of Ξ whose average magnitude is $< \frac{\delta}{10}$ are removed.

Because the time series are all noisy, it is likely that some the differential equations returned by the unregularized SINDy algorithm will acquire spurious terms. Insisting that only terms which *most* of the models find useful are kept, as with our group sparsity method, should help to mitigate this issue. In this way we are able to leverage the fact that we have multiple trials involving similar objects to improve the robustness of the learned models to noise. Even if some of the unregularized models from a given drop involve erroneous library functions, we might still hope that, on average, the models will pick the correct terms. Our approach can also be viewed as a type of *ensemble* method wherein a set of models is formed from the time series of a given drop, they are allowed to vote on which terms are important, then the models are retrained using the constrained set of library functions agreed upon in the previous step.

2.3.3. Equations of Motion

Even the simplest model for the height, $x(t)$, of a falling object involves an acceleration term. Consequently, we impose the restriction that our model be a second order (autonomous) differential equation:

$$\ddot{x} = f(x, \dot{x}). \tag{6}$$

The SINDy framework is designed to work with first order systems of differential equations, so we convert (6) into such a system:

$$\begin{cases} \dot{x} = & v \\ \dot{v} = & g(x, v). \end{cases}$$

We then apply SINDy, with $\mathbf{x} = \begin{bmatrix} x & v \end{bmatrix}^\top$ and $f(\mathbf{x}) = \begin{bmatrix} v & g(\mathbf{x}) \end{bmatrix}^\top$, and attempt to learn the function g. In fact, because we already know the correct right-hand side function for $\dot{x}$, we need only concern ourselves with finding an expression for $\dot{}v$.

Our non-linear library consists of polynomials in x and v up to degree three, visualized in **Figure 5**:

$$\Phi(\mathbf{X}) = \begin{bmatrix} | & | & | & | & | & | & & | \\ \mathbf{1} & \mathbf{x}(t) & \mathbf{v}(t) & \mathbf{x}(t)\mathbf{v}(t) & \mathbf{x}(t)^2 & \mathbf{v}(t)^2 & \ldots & \mathbf{v}(t)^3 \\ | & | & | & | & | & | & & | \end{bmatrix}. \tag{7}$$

Assuming that the motion of the balls is completely determined by Newton's second law, $F = ma = m\ddot{x}$, we may interpret the SINDy algorithm as trying to discover the force (after dividing by mass) that explains the observed acceleration.

Though we know now that the acceleration of a ball should not depend on its height, we seek to place ourselves in a position of ignorance analogous to the position scientists would have found themselves in centuries ago. We leave it to our algorithm to sort out which terms are appropriate. In practice one might selectively choose which functions to include in the library based on domain knowledge, or known properties of the system being modeled.

2.3.4. Numerical Differentiation

In order to form the non-linear library (7) and the derivative matrix, $\dot{\mathbf{X}}$, we must approximate the first two derivatives of the height data from each drop. Applying standard numerical

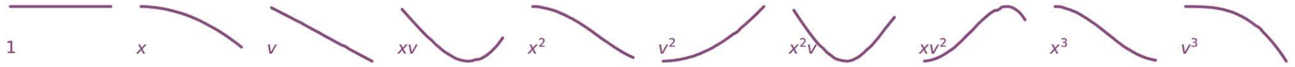

FIGURE 5 | Visualizations of non-linear library functions corresponding to the second green basketball drop. If the motion of the balls is described by Newton's second law, $F = m\ddot{x}$, then these functions can be interpreted as possible forcing terms constituting F.

differentiation techniques to a signal amplifies any noise that is present. This poses a serious problem since we aim to fit a model to the *second* derivative of the height measurements. Because the amount of noise in our data set is non-trivial, two iterations of numerical differentiation will create an intolerable noise level. To mitigate this issue we apply a Savitzky-Golay filter from Savitzky and Golay (1964) to smooth the data before differentiating via second order centered finite differences. Points in a noisy data set are replaced by points lying on low-degree polynomials which are fit to localized patches of the original data with a least-squares method. Other available approaches include using a total variation regularized derivative as in Brunton et al. (2016) or working with an integral formulation of the governing equations as described in Schaeffer and McCalla (2017). We perform a detailed analysis of the error introduced by smoothing and numerical differentiation in the **Supplementary Material**.

3. RESULTS

3.1. Learned Terms

In this section we compare the terms present in the governing equations identified using the unregularized SINDy approach with those present when the group sparsity constraint is imposed. We train separate models on the two drops. The two algorithms are given one sparsity hyperparameter each to be applied for all balls in both drops. The group sparsity method used a value of 1.5 and the other method used a value of 0.04. These parameters were chosen by hand to balance allowing the algorithms enough expressiveness to model the data, while being restrictive enough to prevent widespread overfitting; increasing them produces models with one or no terms and decreasing them results in models with large numbers of terms. See the **Supplementary Material** for a more detailed discussion of our choice of sparsity parameter values.

Figure 6 summarizes the results of this experiment. Learning a separate model for each ball independent of the others allows many models to fall prey to overfitting. Note how most of the governing equations incorporate an extraneous height term. On the other hand, two of the learned models involve only constant acceleration and fail to identify any effect resembling air resistance.

The method leveraging group sparsity is more effective at eliminating extraneous terms and selecting only those which are useful across most balls. Moreover, only the constant and velocity terms are active, matching our intuition that the dominant forces at work are gravity and drag due to air resistance. Interestingly, the method prefers a linear drag term, one proportional to v, to model the discrepancy between measured trajectories and constant acceleration. Even the balls which don't include a velocity term in the unregularized model have this term when group sparsity regularization is employed. This shows that group penalty can simultaneously help to dismiss distracting candidate functions and promote correct terms that may have been overlooked. Is is also reassuring to see that, compared to the other balls, the whiffle ball models have larger coefficients on the v terms. Their accelerations slow at a faster rate as a function of their velocities than do the other balls.

The actual governing equations learned with the group sparsity method are provided in **Table 2**. Every equation has a constant acceleration term within a few meters per second squared of -9.8, but few are quite as close as one might expect. Thus even with a stable method of inferring governing equations, based on this data one would not necessarily conclude that all balls experience the same (mass-divided) force due to gravity. Note also that some of the balls mistakenly adopt *positive* coefficients multiplying v. The balls for which this occurs tend to be those whose motion is well-approximated by constant acceleration. Because the size of the discrepancy between a constant acceleration model and these balls' measured trajectories is not much larger than the amount of error suspected to be present in the data, SINDy has a difficult time choosing an appropriate value for the v terms. One would likely need higher resolution, higher accuracy measurement data in order to obtain reasonable approximations of the drag coefficients or v^2 terms.

At 65°F, the density of air ρ at sea level is 1.211kg/m^3 White and Chul (2011) and its dynamic viscosity μ is $1.82 \times 10^{-5}\text{kg/(m s)}$. The Reynolds number for a ball with diameter D and velocity v will then be

$$Re = 0.667Dv \times 10^5.$$

Table 1 gives the maximum velocities of each ball over the two drops and the corresponding Reynolds numbers. Note that these are the *maximum* Reynolds numbers, not the Reynolds numbers over the entire trajectories. With velocities under 30m/s and diameters from 0.04 to 0.22 m we should expect Reynolds numbers with magnitudes ranging from 10^4 to 10^5 over the course of the balls' trajectories (apart from the very beginnings of each drop). The *average* trajectory consists of about 49 measurements, just over one of which corresponds to a Reynolds number that is $\mathcal{O}\left(10^3\right)$. About 13 of these measurements are associated with Reynolds numbers on the order of 10^4 and roughly 33 with Reynolds numbers of magnitude 10^5. Note that this means the majority of data points were collected when the balls were in the quadratic drag regime. Based on **Figure 1** we should expect balls with Reynolds numbers $< 10^5$ to have drag coefficients of magnitude about 0.5. **Figure 1** suggests that balls experiencing higher Reynolds numbers, such as the volleyball and basketballs should have smaller drag coefficients varying between

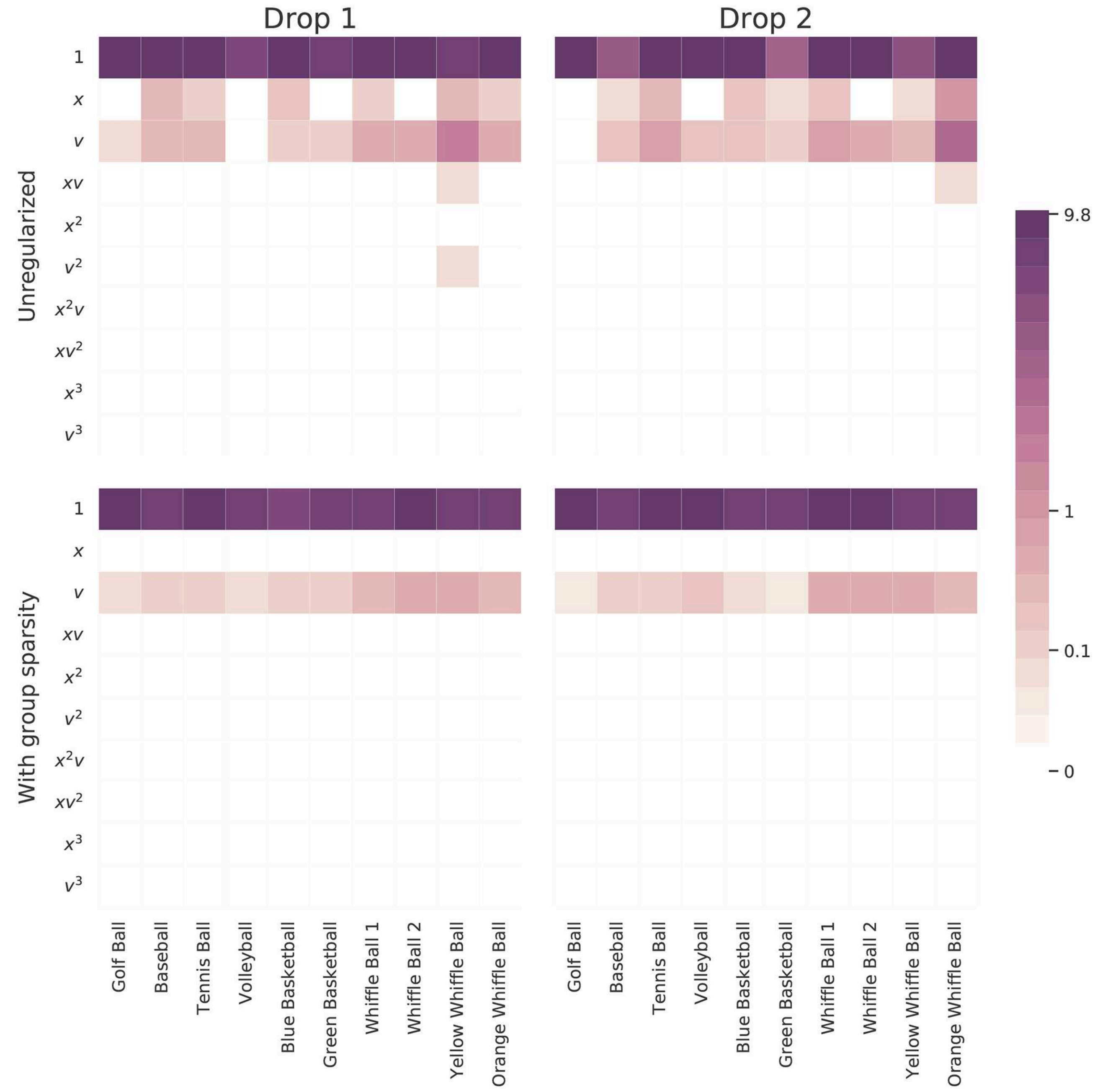

FIGURE 6 | Magnitudes of the coefficients learned for each ball by models trained on one drop either with or without the proposed group sparsity approach. The unregularized approach used a sparsity parameter of 0.04 and the group sparsity method used a value of 1.5. Increasing this parameter slightly in the unregularized case serves to push many models to use only a constant function.

0.05 and 0.3 depending on their smoothness. The predicted (linear) drag coefficients for the volleyball lie in this range while the basketballs' learned drag coefficients are erroneously positive. If the basketballs are treated as being smooth, their drag coefficients predicted by **Figure 1** may be too small for SINDy to identify given the noisy measurement data. A similar effect seems to occur for the golf ball. Though it experiences a lower Reynolds number, its dimples induce a turbulent flow over its surface, granting it a small drag coefficient at a lower Reynolds number. Overall, the linear drag coefficients predicted by the model are at least within a physically reasonable range, with some outliers having incorrect signs.

Next we turn to the simulated data set. We perform the same experiment as with the real world data: we apply both versions of SINDy to a series of simulated ball drops and then note the models that are inferred. Our findings are shown in

TABLE 2 | Models learned by applying SINDy with group sparsity regularization (sparsity parameter $\delta = 1.5$) to each of the two ball drops.

Ball	First drop	Second drop
Golf ball	$\ddot{x} = -9.34 + 0.05v$	$\ddot{x} = -9.44 - 0.03v$
Baseball	$\ddot{x} = -8.51 + 0.14v$	$\ddot{x} = -7.56 + 0.14v$
Tennis ball	$\ddot{x} = -9.08 - 0.13v$	$\ddot{x} = -8.64 - 0.12v$
Volleyball	$\ddot{x} = -8.11 - 0.08v$	$\ddot{x} = -9.64 - 0.23v$
Blue basketball	$\ddot{x} = -6.71 + 0.15v$	$\ddot{x} = -7.50 + 0.07v$
Green basketball	$\ddot{x} = -7.36 + 0.10v$	$\ddot{x} = -8.05 + 0.02v$
Whiffle ball 1	$\ddot{x} = -8.24 - 0.34v$	$\ddot{x} = -9.44 - 0.43v$
Whiffle ball 2	$\ddot{x} = -9.81 - 0.56v$	$\ddot{x} = -9.79 - 0.48v$
Yellow whiffle ball	$\ddot{x} = -8.50 - 0.47v$	$\ddot{x} = -8.45 - 0.46v$
Orange whiffle ball	$\ddot{x} = -7.83 - 0.35v$	$\ddot{x} = -8.03 - 0.42v$

Figure 7. We need not say much about the standard approach: it does a poor job of identifying coherent models for all levels of noise. The group sparsity regularization is much more robust to noise, identifying the correct terms and their magnitudes for noise levels up to half a meter (in standard deviation). For more significant amounts of noise, even this method is unable to decide between adopting x or v into its models. Perhaps surprisingly, if a v^2 term with coefficient $\sim$ 0.1 is added to the simulated model[1], the learned coefficients look nearly identical. Although this additional term visibly alters the trajectory (before it is corrupted by noise), none of the learned equations capture it, even in the absence of noise. One reason for this is because the coefficient multiplying v^2 is too small to be retained during the sequential thresholding least squares procedure. If we decrease the sparsity parameter enough to accommodate it, the models also acquire spurious higher order terms. To infer the v^2 term using the approach outlined here, one would need to design and carry out additional experiments which better isolate this effect, perhaps by using a denser fluid or by dropping a ball with a larger diameter of relatively small mass, thereby increasing the constant multiplying v^2C_D in (1). A much more realistic drag force based on (1) can be used to simulate falling balls. Such a drag force will shift from being linear to quadratic in v over the course of a ball's trajectory. In this scenario neither version of SINDy identifies a v^2 term, regardless of how much many measurements are collected, but both detect linear drag, exhibiting similar performance as is shown here. A more detailed discussion can be found in the **Supplementary Material**.

3.2. Model Error

We now turn to the problem of testing the predictive performance of models learned from the data. We benchmark four models of increasing complexity on the drop data. The model templates are as follows:

1. Constant acceleration: $\ddot{x} = \alpha$
2. Constant acceleration with linear drag: $\ddot{x} = \alpha + \beta v$
3. Constant acceleration with linear and quadratic drag: $\ddot{x} = \alpha + \beta v + \gamma v^2$
4. Overfit model: Set a low sparsity threshold and allow SINDy to fit a more complicated model to the data

The model parameters α, β, and γ are learned using the SINDy algorithm using libraries consisting of just the terms required by the templates. The testing procedure consists of constructing a total of 80 models (4 templates × 10 balls × 2 drops) and then using them to predict a quantity of interest. First a template model is selected then it is trained using one ball's trajectory from one drop. Once trained, the model is given the initial conditions (initial height and velocity) from the same ball's other drop and tasked with predicting the ball's height after 2.8 s have passed[2]. Recall from **Figure 4** that the same ball dropped twice from the same height by the same person on the same day can hit the ground at substantially different times. In the absence of any confounding factors, the time it takes a sphere to reach the ground after being released will vary significantly based on its initial velocity. Since there is sure to be some error in estimating the initial height and velocity of the balls, we should expect only modest accuracy in predicting their landing times. We summarize the outcome of this experiment in **Figure 8**. The error tends to decrease significantly between model one and model two, marking a large step in explaining the discrepancy between a constant acceleration model and observation. There does not appear to be a large difference between the predictive powers of models two and three as both seem to provide similar levels of accuracy. Occam's razor might be invoked here to motivate a preference for model two over model three since it is simpler and has the same accuracy. This provides further evidence that the level of noise and error in the data set is too large to allow one to accurately infer the dynamics due to v^2. Adding additional terms to the equations seems to weaken their generalizability somewhat, as indicated by the slight increase in errors for model four.

Figure 9 visualizes the forecasts of the learned equations for two of the balls along with their deviation from the true measurements. The models are first trained on data from drop 2, then they are given initial conditions from the same drop and made to predict the full trajectories. There are a few observations to be made. The constant acceleration models (model one) are clearly inadequate, especially for the whiffle ball. Their error is much higher than that of the other models indicating that they are underfitting the data, though constant acceleration appears to be a reasonable approximation for a falling golf ball. Models two through four all seem to be imitating the trajectories to about the level of the measurement noise, which is about the most we could hope of them. It is difficult to say which model is best by looking at these plots alone. To break the tie we can observe what happens if we

[1] It should be noted that, based on the balls' approximated velocities, the largest coefficient multiplying v^2 (i.e., $\frac{1}{2m}\rho AC_D$ from (1), where m is the mass of a ball), is < 0.08 in magnitude, across all the trials.

[2] This number corresponds to the shortest set of measurement data across all the trials. All models are evaluated at 2.8 s to allow for meaningful comparison of error rates between models.

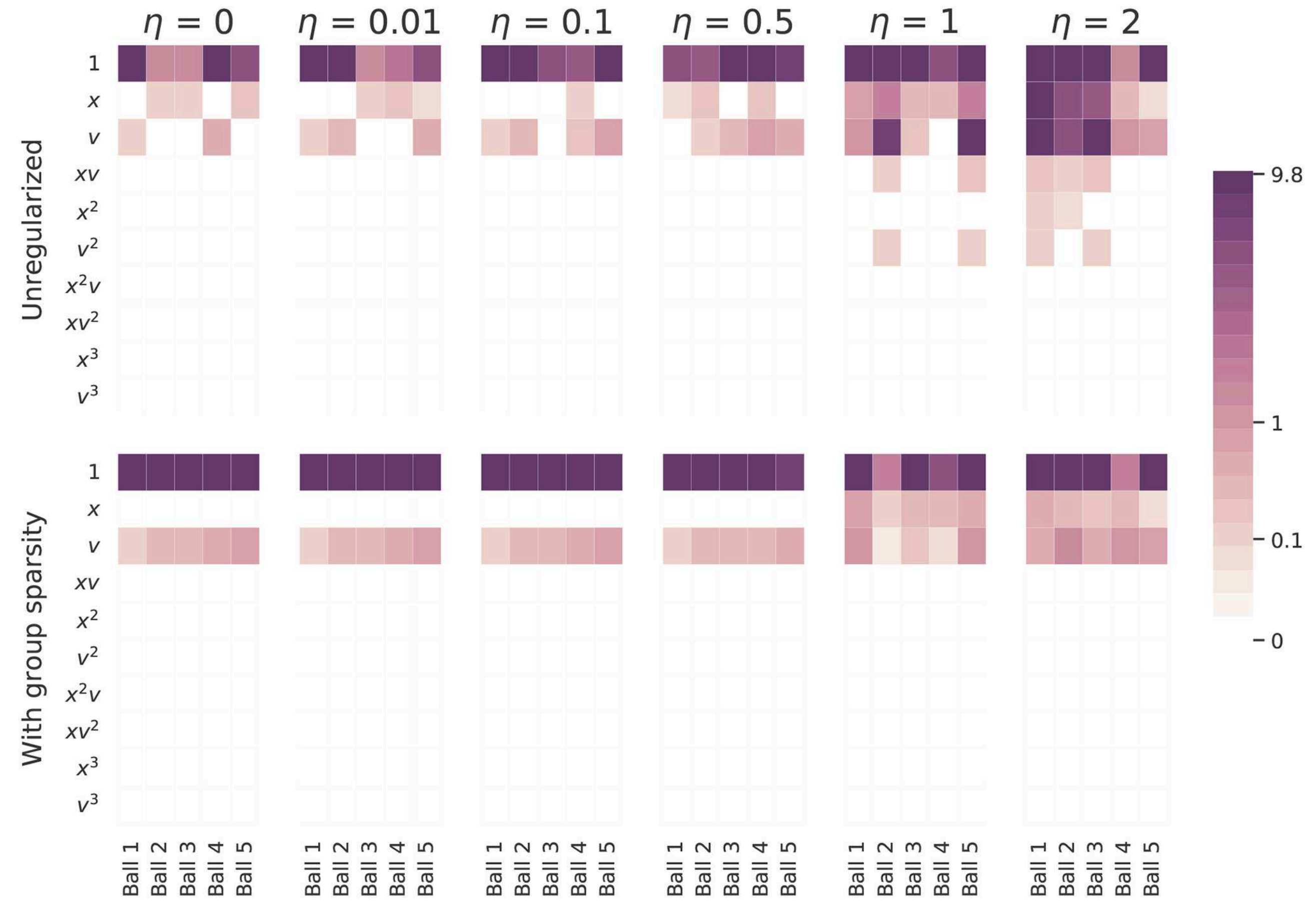

FIGURE 7 | A comparison of coefficients of the models inferred from the simulated falling balls. The top row shows the coefficients learned with the standard SINDy algorithm and the bottom row the coefficients learned with the group sparsity method. η indicates the amount of noise added to the simulated ball drops. The standard approach used a sparsity parameter of 0.05 and the group sparsity method used a value of 1.5. The balls were simulated using constant acceleration and the following respective coefficients multiplying v: −0.1, −0.3, −0.3, −0.5, −0.7.

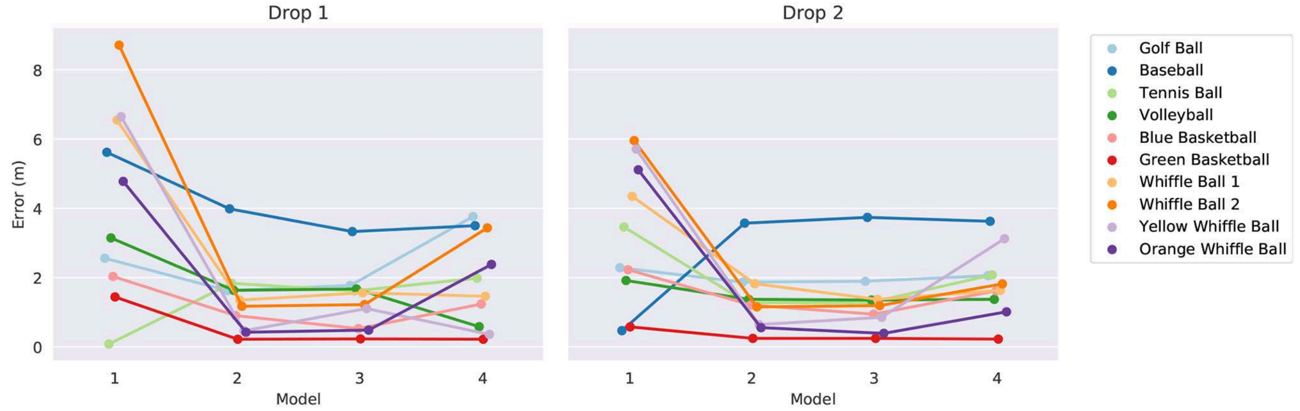

FIGURE 8 | The error in landing time predictions for the four models. The results for the models trained on drops one and two are shown on the left and right, respectively. We have intentionally jittered the horizontal positions of the data points to facilitate easier comparison.

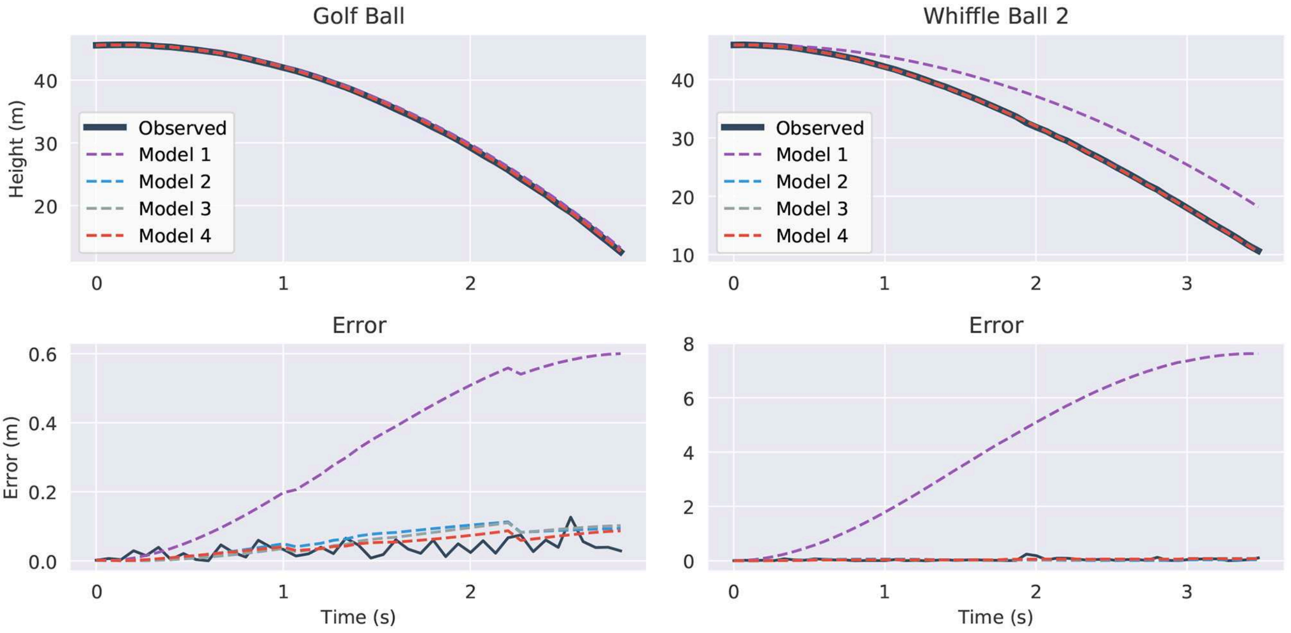

FIGURE 9 | Predicted trajectories and error for the Golf Ball (top) and Whiffle Ball 2 (bottom). On the left we compare the predicted trajectories against the true path and on the right we show the absolute error for the predictions. The "Observed" lines in the error plots show the difference between the original height measurements and the smoothed versions used for differentiation. They give an idea of the amount of intrinsic measurement noise. All models plotted were trained and evaluated on drop 2.

evaluate the models in "unfamiliar" circumstances and force them to extrapolate.

Supplying the same initial conditions as before, with initial height shifted up to avoid negative heights, we task the models with predicting the trajectories out to 15 s. The results are shown in **Figure 10**. All four models fit the observed data itself fairly well. However, 6 or 7 s after the balls are released, a significant degree of separation has started to emerge between the trajectories. The divergence of the model four instances is the most abrupt and the most pronounced. The golf ball's model grows without bound after 7 s. It is here that the danger of overfit, high-order models becomes obvious. In contrast, the other models are better behaved. For the golf ball models one through three agree relatively well, perhaps showing that it is easier to predict the path of a falling golf ball than a falling whiffle ball. That model two is so similar to the constant acceleration of model one also suggests that the golf ball experiences very little drag. The v^2 term for model three has a coefficient which is erroneously positive and essentially cancels out the speed dampening effects of the drag term, leading to an overly rapid predicted descent. Models two and three agree extremely well for the whiffle ball as the learned v^2 coefficient is very small in magnitude.

4. DISCUSSION AND CONCLUSIONS

In this work, we have revisited the classic problem of modeling the motion of falling objects in the context of modern machine learning, sparse optimization, and model selection. In particular, we develop data-driven models from experimental position measurements for several falling spheres of different size, mass, roughness, and porosity. Based on this data, a hierarchy of models are selected via sparse regression in a library of candidate functions that may explain the observed acceleration behavior. We find that models developed for individual ball-drop trajectories tend to overfit the data, with all models including a spurious height-dependent force and lower-density balls resulting in additional spurious terms. Next, we impose the assumption that all balls must be governed by the same basic model terms, perhaps with different coefficients, by considering all trajectories simultaneously and selecting models via group sparsity. These models are all parsimonious, with only two dominant terms, and they tend to generalize without overfitting.

Although we often view the motion of falling spheres as a solved problem, the observed data is quite rich, exhibiting a range of behaviors. In fact, a constant gravitational acceleration is not immediately obvious, as the falling motion is strongly affected by complex unsteady fluid drag forces; the data alone would suggest that each ball has its own slightly different gravity constant. It is interesting to note that our group sparsity models include a drag force that is proportional to the velocity, as opposed to the *textbook* model that includes the square of velocity that is predicted for a constant drag coefficient. However, in reality the drag coefficient decreases with velocity, as shown in **Figure 1**, which may contribute to the force being proportional to velocity. Even when a higher fidelity drag model is used—a model containing rational terms missing from and poorly

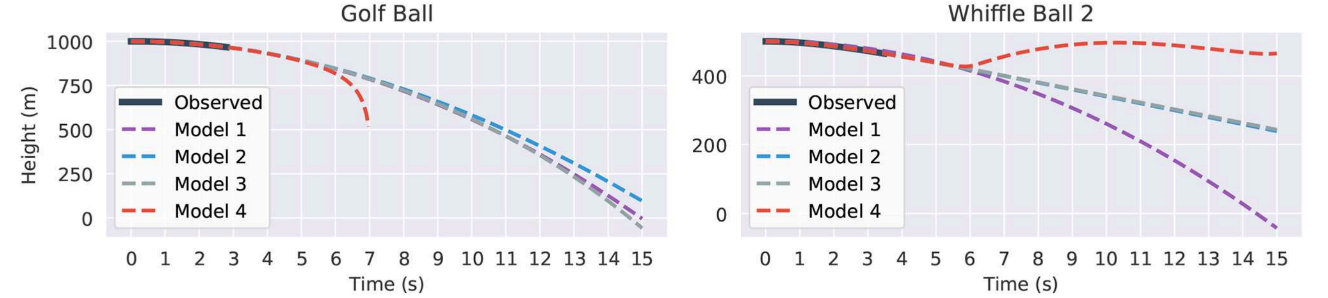

FIGURE 10 | Fifteen seconds forecasted trajectories for the Golf Ball (left) and Whiffle Ball 2 (right) based on the second drop. Part of the graph of Model 4 (red) is omitted in the Golf Ball plot because it diverged to $-\infty$.

approximated by the polynomial library functions—to collect measurements uncorrupted by noise, SINDy struggles to identify coherent dynamics. In general SINDy may not exhibit optimal performance if not equipped with a library of functions in which dynamics can be represented sparsely. We emphasize that although the learned models tend to fit the data relatively well, it would be a mistake to assume that they would retain their accuracy for Reynolds numbers larger than those present in the training data. In particular we should expect the models to have trouble extrapolating beyond the drag crisis where the dynamics change considerably. This weakness is inherent in virtually all machine learning models; their performance is best when they are applied to data similar to what they have already seen and dubious when applied in novel contexts. That is to say they excel at interpolation, but are often poor extrapolators.

Collecting a richer set of data should enable the development of refined models with more accurate drag physics[3], and this is the subject of future work. In particular, it would be interesting to collect data for spheres falling from greater heights, so that they reach terminal velocity. It would also be interesting to systematically vary the radius, mass, surface roughness, and porosity, for example to determine non-dimensional parameters. Finally, performing similar tests in other fluids, such as water, may also enable the discovery of added mass forces, which are quite small in air. Such a dataset would provide a challenging motivation for future machine learning techniques.

We were able to draw upon previous fluid dynamics research to establish a "ground truth" model against which to compare the models proposed by SINDy. However, in less mature application areas one may not be fortunate enough to have a theory-backed set of reference equations, making it challenging to assess the quality of learned models. Many methods in numerical analysis come equipped with *a priori* or *a posteriori* error estimators or convergence results to give one an idea of the size of approximation errors. Similarly, in statistics goodness of fit estimators exist to help guide practitioners about what type of performance they should expect from various models. A comprehensive study into whether similar techniques could be adopted for application to SINDy would be an interesting topic for future research efforts.

We believe that it is important to draw a parallel between great historical scientific breakthroughs, such as the discovery of a universal gravitational constant, and modern approaches in machine learning. Although computational learning algorithms are becoming increasingly powerful, they face many of the same challenges that human scientists have faced for centuries. These challenges include trade offs between model fidelity and the quality and quantity of data, with inaccurate measurements degrading our ability to disambiguate various physical effects. With noisy data, one can only expect model identification techniques to uncover the dominant, leading-order effects, such as gravity and simple drag; for subtler effects, more accurate measurement data is required. Modern learning architectures are often also prone to overfitting without careful cross-validation and regularization, and models that are both interpretable and generalizable come at a premium. Typically the regularization encodes some basic human assumption, such as sparse regularization, which promotes parsimony in models. More fundamentally, it is not always clear what should be measured, what terms should be modeled, and what parameters should be varied to isolate the effect one wishes to study. Historically, this type of scientific inquiry has been driven by human curiosity and intuition, which will be critical elements if machine intelligence is to advance scientific discovery.

DATA AVAILABILITY STATEMENT

The code and datasets used in this study are available at https://github.com/briandesilva/discovery-of-physics-from-data.

AUTHOR CONTRIBUTIONS

BS: implementation, numerical experiments, results analysis, and manuscript writing. DH: falling object experiments and scientific advice.

[3] We note that in order to properly resolve these more complex drag dynamics with SINDy the candidate library would likely need to be enriched.

SB and JK: scientific advice, results analysis, and manuscript writing.

ACKNOWLEDGMENTS

We would like to acknowledge funding support from the Defense Advanced Research Projects Agency (DARPA PA-18-01-FP-125) and the Air Force Office of Scientific Research (FA9550-18-1-0200 for SB and FA9550-17-1-0329 for JK).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frai.2020.00025/full#supplementary-material

REFERENCES

Achenbach, E. (1972). Experiments on the flow past spheres at very high reynolds numbers. *J. Fluid Mech.* 54, 565–75. doi: 10.1017/S0022112072000874

Achenbach, E. (1974). Vortex shedding from spheres. *J. Fluid Mech.* 62, 209–21. doi: 10.1017/S0022112074000644

Adler, C. G., and Coulter, B. L. (1978). Galileo and the tower of pisa experiment. *Am. J. Phys.* 46, 199–201. doi: 10.1119/1.11165

Bartlett, D., Goldhagen, P., and Phillips, E. (1970). Experimental test of coulomb's law. *Phys. Rev. D* 2:483. doi: 10.1103/PhysRevD.2.483

Battaglia, P., Pascanu, R., Lai, M., Rezende, D. J., and Kavukcuoglu, K. (2016). "Interaction networks for learning about objects, relations and physics," in *Advances in Neural Information Processing Systems*, (Long Beach, CA), 4502–10.

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., et al. (2018). Relational inductive biases, deep learning, and graph networks. *arXiv* 1806.01261.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. New York, NY: Springer.

Blumer, A., Ehrenfeucht, A., Haussler, D., and Warmuth, M. K. (1987). Occam's razor. *Inform. Process. Lett.* 24, 377–380. doi: 10.1016/0020-0190(87)90114-1

Bongard, J., and Lipson, H. (2007). Automated reverse engineering of nonlinear dynamical systems. *Proc. Natl. Acad. Sci. U.S.A.* 104, 9943–9948. doi: 10.1073/pnas.0609476104

Breiman, L. (2001). Statistical modeling: the two cultures (with comments and a rejoinder by the author). *Stat. Sci.* 16, 199–231. doi: 10.1214/ss/1009213726

Bridewell, W., Langley, P., Todorovski, L., and Džeroski, S. (2008). Inductive process modeling. *Mach. Learn.* 71, 1–32. doi: 10.1007/s10994-007-5042-6

Brunton, S. L., Proctor, J. L., and Kutz, J. N. (2016). Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc. Natl. Acad. Sci. U.S.A.* 113, 3932–3937. doi: 10.1073/pnas.1517384113

Calvert, J. (1972). Some experiments on the flow past a sphere. *Aeronaut. J.* 76, 248–250.

Champion, K., Zheng, P., Aravkin, A. Y., Brunton, S. L., and Kutz, J. N. (2019). A unified sparse optimization framework to learn parsimonious physics-informed models from data. *arXiv* 1906.10612.

Chang, M. B., Ullman, T., Torralba, A., and Tenenbaum, J. B. (2016). A compositional object-based approach to learning physical dynamics. *arXiv* 1612.00341.

Christensen, R. S., Teiwes, R., Petersen, S. V., Uggerhøj, U. I., and Jacoby, B. (2014). Laboratory test of the galilean universality of the free fall experiment. *Phys. Educ.* 49:201. doi: 10.1088/0031-9120/49/2/201

Cooper, L. (1936). *Aristotle, Galileo, and the Tower of Pisa*, New York, NY: Cornell University Press Ithaca.

Cross, R., and Lindsey, C. (2014). Measuring the drag force on a falling ball. *Phys. Teacher* 52, 169–170. doi: 10.1119/1.4865522

Dam, M., Brøns, M., Juul Rasmussen, J., Naulin, V., and Hesthaven, J. S. (2017). Sparse identification of a predator-prey system from simulation data of a convection model. *Phys. Plasmas* 24:022310. doi: 10.1063/1.4977057

Domingos, P. (1999). The role of occam's razor in knowledge discovery. *Data Mining Knowl. Discov.* 3, 409–425. doi: 10.1023/A:1009868929893

Falconer, I. (2017). No actual measurement…was required: Maxwell and cavendish's null method for the inverse square law of electrostatics. *Stud. Hist. Philos. Sci. A* 65, 74–86. doi: 10.1016/j.shpsa.2017.05.001

Goff, J. E. (2013). A review of recent research into aerodynamics of sport projectiles. *Sports Eng.* 16, 137–154. doi: 10.1007/s12283-013-0117-z

Hoffmann, M., Fröhner, C., and Noé,, F. (2019). Reactive SINDy: discovering governing reactions from concentration data. *J. Chem. Phys.* 150:025101. doi: 10.1063/1.5066099

Kaewsutthi, C., and Wattanakasiwich, P. (2011). "Student learning experiences from drag experiments using high-speed video analysis," in *Proceedings of The Australian Conference on Science and Mathematics Education (formerly UniServe Science Conference)*, (Melbourne), Vol. 17.

Kaiser, E., Kutz, J. N., and Brunton, S. L. (2018). Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proc. R. Soc. Lond. A* 474:2219. doi: 10.1098/rspa.2018.0335

Kepler, J. (2015). *Astronomia Nova. Pragae 1609*. Green Lion Press.

Lai, Z., and Nagarajaiah, S. (2019). Sparse structural system identification method for nonlinear dynamic systems with hysteresis/inelastic behavior. *Mech. Syst. Signal Process.* 117, 813–42. doi: 10.1016/j.ymssp.2018.08.033

Loiseau, J.-C., and Brunton, S. L. (2018). Constrained sparse Galerkin regression. *J. Fluid Mech.* 838, 42–67. doi: 10.1017/jfm.2017.823

Loiseau, J.-C., Noack, B. R., and Brunton, S. L. (2018). Sparse reduced-order modeling: sensor-based dynamics to full-state estimation. *J. Fluid Mech.* 844, 459–90. doi: 10.1017/jfm.2018.147

Magarvey, R., and MacLatchy, C. (1965). Vortices in sphere wakes. *Can. J. Phys.* 43, 1649–56. doi: 10.1139/p65-154

Mangan, N. M., Brunton, S. L., Proctor, J. L., and Kutz, J. N. (2016). Inferring biological networks by sparse identification of nonlinear dynamics. *IEEE Trans. Mol. Biol. Multiscale Commun.* 2, 52–63. doi: 10.1109/TMBMC.2016.2633265

Mangan, N. M., Kutz, J. N., Brunton, S. L., and Proctor, J. L. (2017). Model selection for dynamical systems via sparse regression and information criteria. *Proc. R. Soc. A* 473, 1–16. doi: 10.1098/rspa.2017.0009

Maxwell, J. C. (1873). *A Treatise on Electricity and Magnetism*, Vol. 1. Oxford: Clarendon Press.

Mehta, R. D. (1985). Aerodynamics of sports balls. *Annu. Rev. Fluid Mech.* 17, 151–189. doi: 10.1146/annurev.fl.17.010185.001055

Mehta, R. D. (2008). "Sports ball aerodynamics," in *Sport Aerodynamics* (Vienna: Springer), 229–331. doi: 10.1007/978-3-211-89297-8_12

Moller, W. (1938). Experimentelle untersuchung zur hydromechanick der hugel. *Phys. Z.* 35, 57–80.

Mrowca, D., Zhuang, C., Wang, E., Haber, N., Fei-Fei, L. F., Tenenbaum, J., et al. (2018). "Flexible neural representation for physics prediction," in *Advances in Neural Information Processing Systems*, (Montréal), 8799–8810.

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. Cambridge, MA: MIT Press.

Newman, J. (1977). *Marine Hydrodynamics*. Cambridge, MA: The MIT Press.

Newton, I. (1999). *The Principia: Mathematical Principles of Natural Philosophy*. Berkeley, CA: University of California Press.

Owen, J. P., and Ryu, W. S. (2005). The effects of linear and quadratic drag on falling spheres: an undergraduate laboratory. *Eur. J. Phys.* 26:1085. doi: 10.1088/0143-0807/26/6/016

Peters, C. H. F., and Knobel, E. B. (1915). *Ptolemy's Catalogue of Stars: A Revision of the Almagest*. Washington, DC: Physics and Chemistry in Space.

Ptolemy, C. (2014). *The Almagest: Introduction to the Mathematics of the Heavens*. Santa Fe, NM: Green Lion Press.

Raissi, M. (2018). Deep hidden physics models: Deep learning of nonlinear partial differential equations. *J. Mach. Learn. Res.* 19, 932–55. Available online at: http://jmlr.org/papers/v19/18-046.html.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2017a). Machine learning of linear differential equations using gaussian processes. *J. Comput. Phys.* 348, 683–93. doi: 10.1016/j.jcp.2017.07.050

Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2017b). Physics informed deep learning (part I): data-driven solutions of nonlinear partial differential equations. *arXiv* 1711.10561.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2017c). Physics informed deep learning (part II): data-driven discovery of nonlinear partial differential equations. *arXiv* 1711.10566.

Rudy, S., Alla, A., Brunton, S. L., and Kutz, J. N. (2019). Data-driven identification of parametric partial differential equations. *SIAM J. Appl. Dyn. Syst.* 18, 643–60. doi: 10.1137/18M1191944

Rudy, S. H., Brunton, S. L., Proctor, J. L., and Kutz, J. N. (2017). Data-driven discovery of partial differential equations. *Sci. Adv.* 3:e1602614. doi: 10.1126/sciadv.1602614

Savitzky, A., and Golay, M. J. E. (1964). Smoothing and differentiation of data by simplified least squares procedures. *Anal. Chem.* 36, 1627–39. doi: 10.1021/ac60214a047

Schaeffer, H. (2017). Learning partial differential equations via data discovery and sparse optimization. *Proc. R. Soc. A* 473:20160446. doi: 10.1098/rspa.2016.0446

Schaeffer, H., and McCalla, S. G. (2017). Sparse model selection via integral terms. *Phys. Rev. E* 96:023302. doi: 10.1103/PhysRevE.96.023302

Schaeffer, H., Tran, G., and Ward, R. (2018). Extracting sparse high-dimensional dynamics from limited data. *SIAM J. Appl. Math.* 78, 3279–95. doi: 10.1137/18M116798X

Schmidt, M., and Lipson, H. (2009). Distilling free-form natural laws from experimental data. *Science* 324, 81–85. doi: 10.1126/science.1165893

Segre, M. (1980). The role of experiment in galileo's physics. *Archiv. Hist. Exact Sci.* 23, 227–252. doi: 10.1007/BF00357045

Smits, A. J., and Ogg, S. (2004). "Aerodynamics of the golf ball," in *Biomedical Engineering Principles in Sports* (New York, NY: Springer), 3–27. doi: 10.1007/978-1-4419-8887-4_1

Sorokina, M., Sygletos, S., and Turitsyn, S. (2016). Sparse identification for nonlinear optical communication systems: SINO method. *Opt. Express* 24, 30433–30443. doi: 10.1364/OE.24.030433

Su, W., Bogdan, M., and Candes, E. (2017). False discoveries occur early on the lasso path. *Ann. Stat.* 45, 2133–2150. doi: 10.1214/16-AOS1521

Sznitman, J., Stone, H. A., Smits, A. J., and Grotberg, J. B. (2017). Teaching the falling ball problem with dimensional analysis. *Eur. J. Phys. Educ.* 4, 44–54.

Tanevski, J., Simidjievski, N., Todorovski, L., and Džeroski, S. (2017). "Process-based modeling and design of dynamical systems," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (Springer), 378–382. doi: 10.1007/978-3-319-71273-4_35

Tanevski, J., Todorovski, L., and Džeroski, S. (2016). Learning stochastic process-based models of dynamical systems from knowledge and data. *BMC Syst. Biol.* 10:30. doi: 10.1186/s12918-016-0273-4

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. B Methodol.* 58, 267–288. doi: 10.1111/j.2517-6161.1996.tb02080.x

Tran, G., and Ward, R. (2016). Exact recovery of chaotic systems from highly corrupted data. *arXiv* 1607.01067.

Watters, N., Zoran, D., Weber, T., Battaglia, P., Pascanu, R., and Tacchetti, A. (2017). "Visual interaction networks: learning a physics simulator from video," in *Advances in Neural Information Processing Systems*, (Long Beach, CA), 4539–4547.

White, F. M., and Chul, R. (2011). *Fluid Mechanics, 2011*. New-York, NY: MacGraw-Hill.

Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., et al. (2008). Top 10 algorithms in data mining. *Knowl. Inform. Syst.* 14, 1–37. doi: 10.1007/s10115-007-0114-2

Yuan, M., and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *J. R. Stat. Soc. B Stat. Methodol.* 68, 49–67. doi: 10.1111/j.1467-9868.2005.00532.x

Zheng, P., Askham, T., Brunton, S. L., Kutz, J. N., and Aravkin, A. Y. (2018). A unified framework for sparse relaxed regularized regression: Sr3. *IEEE Access* 7, 1404–423. doi: 10.1109/ACCESS.2018.2886528

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.