Symbolic pregression: Discovering physical laws from distorted video

Silviu-Marian Udrescu* and Max Tegmark

Department of Physics, Institute for AI & Fundamental Interactions, and Center for Brains, Minds, & Machines, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA

(Received 11 September 2020; accepted 31 March 2021; published 22 April 2021)

We present a method for unsupervised learning of equations of motion for objects in raw and optionally distorted unlabeled synthetic video (or, more generally, for discovering and modeling predictable features in time-series data). We first train an autoencoder that maps each video frame into a low-dimensional latent space where the laws of motion are as simple as possible, by minimizing a combination of nonlinearity, acceleration, and prediction error. Differential equations describing the motion are then discovered using Pareto-optimal symbolic regression. We find that our pre-regression ("pregression") step is able to rediscover Cartesian coordinates of unlabeled moving objects even when the video is distorted by a generalized lens. Using intuition from multidimensional knot theory, we find that the pregression step is facilitated by first adding extra latent space dimensions to avoid topological problems during training and then removing these extra dimensions via principal component analysis. An inertial frame is autodiscovered by minimizing the combined equation complexity for multiple experiments.

DOI: 10.1103/PhysRevE.103.043307

I. INTRODUCTION

A central goal of physics and science more broadly is to discover mathematical patterns in data. For example, after four years of analyzing data tables on planetary orbits, Kepler started a scientific revolution in 1605 by discovering that Mars's orbit was an ellipse [1]. There has been great recent progress in automating such tasks with *symbolic regression*: discovery of a symbolic expression that accurately matches a given data set [2–23]. Open-source software now exists that can discover quite complex physics equations by combining neural networks with techniques inspired by physics and information theory [22,23].

However, symbolic regression problems are of course just a small subset of the problems scientists face. In this paper, we focus on a different but closely related problem: how to decide which parameters of the observed data we should try to describe with equations. Wigner famously stated that "the world is very complicated and ... the complications are called initial conditions, the domains of regularity, laws of nature" [24], so how can the discovery of these regularities be automated? In Fig. 1, how can an unsupervised algorithm learn that to predict the next video frame, it should focus on the x and y coordinates of the rocket, not on its color or on the objects in the background? More generally, given an evolving data vector with N degrees of freedom, how can we autodiscover which $n < N$ degrees of freedom are most useful for prediction? Renormalization addresses this question in a particular context, but we are interested in generalizing this.

Suppose, for example, that we tried to rediscover Kepler's results by mounting a camera with a wide-angle lens in a dark cloudless location, taking a digital snapshot of the sky at the same sidereal time every night, so that distant stars appeared unmoving. How could a computer algorithm presented with a series of images with say $N = 10^7$ pixels automatically learn that the most useful degrees of freedom for prediction are the position coordinates of the Moon and the visible planets (which Brahe carefully measured and tabulated), not the blackness of the sky, the positions of stars, the color of Mars, or the shape of the Moon?

The goal of our paper is to tackle this pre-regression problem, which we will refer to as "pregression" for brevity. Automated pregression enables laws of motion to be discovered starting with raw observational data such as videos or other time-series data. This can be viewed as a small step toward a particular type of unsupervised learning of physics, whereby an algorithm learns from raw observational data how to predict the future from the past without any human supervision or prior knowledge [25–27].

There has been impressive recent progress on using neural networks for video prediction [28–42] and more general physics problems [27,43–48]. However, these machine-learned models tend to be inscrutable black boxes that provide their human users with limited understanding. In contrast, the machine learning approach in this paper aspires to *intelligible intelligence*, i.e., learning a model of the system that is simple enough for a human user to understand. Such intelligibility (pursued in, e.g., [26,27,49–52]) is a central goal of physics research, and has two advantages:

(1) Understanding how a model works enables us to trust it more, which is particularly valuable when AI systems make decisions affecting people's lives [53–56].

(2) Simple intelligible models such as the laws of physics tend to yield more accurate and generalizable predictions than black-box over-parametrized fits, especially over long timescales. This is why spacecraft navigation systems use Newton's law of gravitation rather than a neural-network-based approximation thereof.

*sudrescu@mit.edu

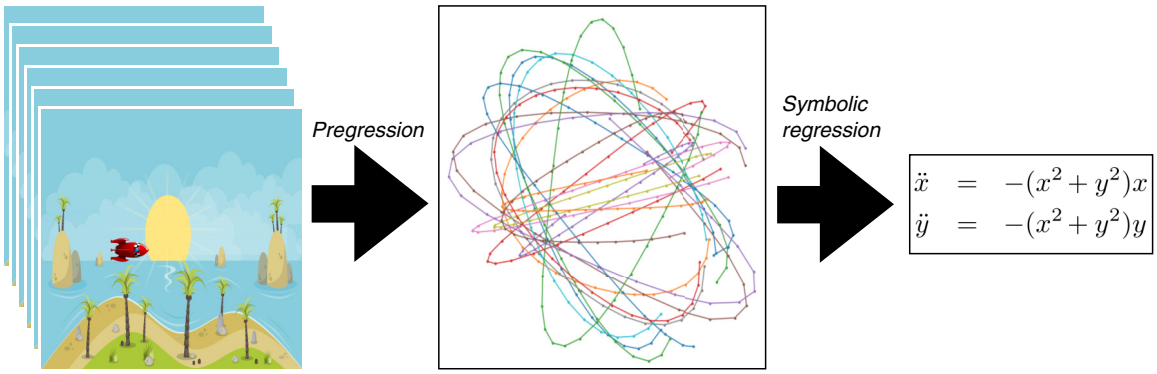

FIG. 1. Our pregression algorithm seeks to autoencode a sequence of video frames (left) corresponding to a specific type of motion into a low-dimensional latent space (middle) where the laws of motion (right) are as simple as possible, in this example those of a quartic oscillator. In the middle figure each point corresponds to the x and y of the rocket in a given frame, while points having the same color or shading and being connected by a line belong to the same trajectory.

The video prediction papers most closely related to the present work take one of two approaches. Some improve accuracy and intelligibility by hardcoding coordinate-finding or physics elements by hand to help learn, e.g., rigid-body motion [57], physical object properties, or partial differential equations [58,59]. The alternative *tabula rasa* approach assumes no physics whatsoever and attempts to learn physical object properties [60], object positions [61,62], object relations [63], and time evolution [64–66] by learning a low-dimensional representation or latent space which is unfortunately too complex or inscrutable to allow discovery of exact equations of motion. The present paper builds on this *tabula rasa* approach; our key contribution is to automatically simplify the latent space, using ideas inspired by general relativity and knot theory, to make the dynamics simple enough for symbolic regression to discover equations of motions.

The rest of this paper is organized as follows. In Sec. II, we present our algorithm. In Sec. III, we test it on simulated videos (such as the flying rocket example in Fig. 1) for motion in a force-free environment, a gravitational field, a magnetic field, a harmonic potential, and a quartic potential. We also test the effects of adding noise and geometric image distortion. We summarize our conclusions and discuss future challenges in Sec. IV.

II. METHOD

The goal of our method is to start with raw video sequences of an object moving in front of some static background, and to, in a fully unsupervised manner (with no input besides the raw video), discover the differential equation governing the object's motion. Our algorithm consists of two parts:

(1) a neural-network-based pregression step that learns to map images into a low-dimensional latent space representing the physically relevant parameters (degrees of freedom), and

(2) a symbolic regression step that discovers the law of motion, i.e., the differential equation governing the time evolution of these parameters.

A. Learning the latent space

Abstractly, we can consider each video frame as a single point in an N-dimensional space, where N is the number of color channels (3 in our case) times the number of pixels in each image. If the motion involves only $n \ll N$ degrees of freedom (for example, $n = 2$ for a rigid object moving without rotating in two dimensions), then all observed points in the N-dimensional space lie on some n-dimensional submanifold that we wish to discover, parametrized by an n-dimensional parameter vector that we can consider as a point in an n-dimensional latent space. Our neural network architecture for learning the latent space is shown in Fig. 2, and consists of three separate feedforward neural networks:

(1) an encoder E that maps images $\mathbf{x}_i \in \mathbb{R}^N$ into latent space vectors $\mathbf{z}_i \in \mathbb{R}^n$,

(2) a decoder D that maps latent space vectors $\mathbf{z}_i$ into images $\mathbf{x}_i$, and

(3) an evolution operator U that predicts the next latent space vector $\mathbf{z}_i$ from the two previous ones (two are needed to infer velocities).[1]

The encoder-decoder pair forms an autoencoder [67–75] that tries to discover the n most dynamically relevant parameters from each movie frame, from which it can be reconstructed as accurately as possible.

B. Quantifying simplicity

It is tempting to view the results of our pregression algorithm as rather trivial, merely learning to extract x and y coordinates of objects. This would be incorrect, however, since we will see that the pregression rediscovers simple physical laws even from video images that are severely warped, as illustrated in Fig. 3, where the learned latent space is a complicated nonlinear function of the Cartesian coordinates. The basic reason for this is that Fig. 2 makes no mention of any preferred latent-space coordinate system. This reparametrization invariance is a double-edged sword, however: a core challenge that we must overcome is that even if the system *can* be described by a simple time evolution U, the basic

[1]The two last images are needed because the laws of physics are second-order differential equations that can be transformed into second-order difference equations; our method trivially generalizes to using the last T inputs for any choice $T = 1, 2, 3, \ldots$.

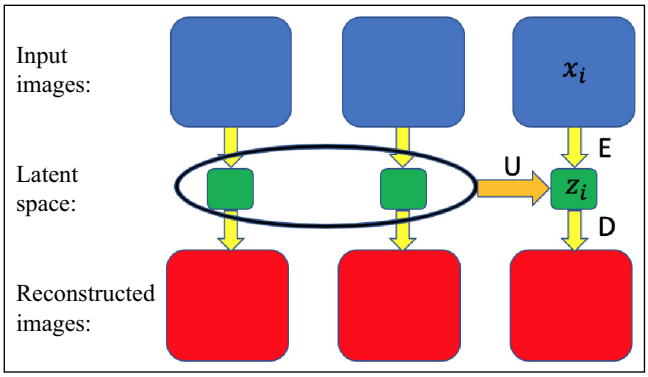

FIG. 2. Architecture of our neural network: An encoder E that maps images $\mathbf{x}_i$ into latent space vectors $\mathbf{z}_i$, a decoder D that maps latent space vectors $\mathbf{z}_i$ back into images $\mathbf{x}_i$, and an evolution operator U that predicts the next latent space vector from the two previous ones.

architecture in Fig. 2 may discover something much more complicated. To see this, suppose that there is an autoencoder (E, D) and evolution operator U providing perfect image reconstruction and prediction, i.e., satisfying

$$\begin{aligned} D(E(\mathbf{x}_i)) &= \mathbf{x}_i, \\ U(\mathbf{z}_{i-2}, \mathbf{z}_{i-1}) &= \mathbf{z}_i, \end{aligned} \tag{1}$$

and that U is a fairly simple function. If we now deform the latent space by replacing $\mathbf{z}$ by $\mathbf{z}' \equiv f(\mathbf{z})$ for some invertible but horribly complicated function f, then it is easy to see that the new mappings defined by

$$\begin{aligned} E'(\mathbf{x}) &\equiv f(E(\mathbf{x})), \\ D'(\mathbf{z}') &\equiv D(f^{-1}(\mathbf{z}')), \\ U'(\mathbf{z}') &\equiv f(U(f^{-1}(\mathbf{z}'))) \end{aligned} \tag{2}$$

will still provide perfect autoencoding and evolution

$$\begin{aligned} D'(E'(\mathbf{x}_i)) &= \mathbf{x}_i, \\ U'(\mathbf{z}_{i-2}, \mathbf{z}_{i-1}) &= \mathbf{z}_i \end{aligned} \tag{3}$$

even though the new evolution operator U' is now very complicated. So in contrast with general relativity where the equations of motion remain formally invariant under reparametrization, here they do not.

Not only *can* our architecture discover unnecessarily complicated solutions, but it by default *will*. We jocularly termed

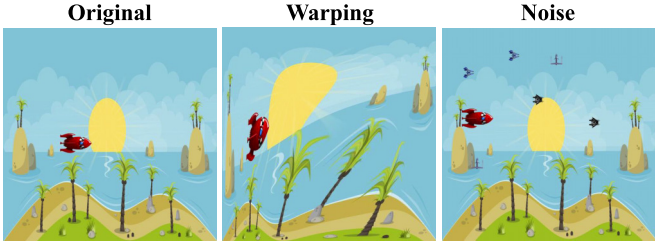

FIG. 3. Our method can rediscover simple laws of motion even if the true images (left) are severely warped (middle) or corrupted by superimposed noise in the form of smaller distractor rockets (right).

this the *Alexander principle* in honor of a child of one of the authors whose sense of humor dictated that he comply with requests in the most complicated way consistent with the instructions. We will face multiple challenges of this type throughout this paper, where our neural networks appeared humorously spiteful simply because they statistically find the most *generic* solution in a vast class of equally accurate ones.

To tackle this problem, we wish to add a regularization term to the loss function that somehow rewards simplicity and penalizes complexity, ideally in a way that involves as few assumptions as possible about the type of dynamics occurring in the video. Defining the $2n$-dimensional vector

$$\mathbf{w}_i \equiv \begin{pmatrix} \mathbf{z}_{i-2} \\ \mathbf{z}_{i-1} \end{pmatrix} \in \mathbb{R}^{2n}, \tag{4}$$

we can view the evolution function $U(\mathbf{w})$ as a mapping from $\mathbb{R}^{2n}$ to $\mathbb{R}^n$ that we wish to be as simple as possible. A natural physics-inspired complexity measure for U is its *curvature*

$$\mathcal{L}^{\text{curv}} \equiv R^{\alpha}_{\mu\nu\beta} R^{\mu\nu\beta}_{\alpha}, \tag{5}$$

defined as the squared Riemann tensor that is ubiquitous in differential geometry and general relativity, defined as

$$R^{\alpha}_{\mu\nu\beta} \equiv \Gamma^{\alpha}_{\nu\beta,\mu} - \Gamma^{\alpha}_{\mu\beta,\nu} + \Gamma^{\gamma}_{\mu\beta}\Gamma^{\alpha}_{\nu\gamma} - \Gamma^{\gamma}_{\nu\beta}\Gamma^{\alpha}_{\mu\gamma}, \tag{6}$$

$$\Gamma^{\alpha}_{\mu\nu} \equiv \tfrac{1}{2} g^{\alpha\sigma}(g_{\sigma\mu,\nu} + g_{\sigma\nu,\mu} - g_{\mu\nu,\sigma}), \tag{7}$$

$$\mathbf{g} \equiv \mathbf{J}\mathbf{J}^t, \tag{8}$$

where $\mathbf{J}$ is the Jacobian of U, the matrix $\mathbf{g}$ is the induced metric on the latent space $\mathbb{R}^n$, indices are raised by multiplying by $\mathbf{g}^{-1}$, commas denote derivatives as in standard tensor notation, and the Einstein summation convention is used. Natural alternatives are the squared Ricci curvature $R^{\mu\nu}R_{\mu\nu}$ or the scalar curvature $R \equiv g^{\mu\nu}R_{\mu\nu}$, where $R_{\mu\nu} \equiv R^{\alpha}_{\mu\alpha\nu}$.

Unfortunately, these curvature measures are numerically cumbersome, since they require taking 3rd derivatives of the neural-network-defined function U and the Riemann tensor has n^4 components. Fortunately, we find that a simpler measure of complexity performs quite well in practice, as reflected by the following loss function:

$$\mathcal{L} \equiv \mathcal{L}^{\text{recon}} + \alpha\mathcal{L}^{\text{pred}} + \beta\mathcal{L}^{\text{nl}} + \gamma\mathcal{L}^{\text{acc}}. \tag{9}$$

These four terms are averages over all time steps i of the following dimensionless functions:

$$\mathcal{L}^{\text{recon}}_i \equiv \frac{|\mathbf{x}_i - D(E(\mathbf{x}_i))|}{|\mathbf{x}_i|}, \tag{10}$$

$$\mathcal{L}^{\text{pred}}_i \equiv \frac{|\mathbf{z}_i - U(\mathbf{z}_{i-2}, \mathbf{z}_{i-1})|}{|\mathbf{z}_{i-1} - \mathbf{z}_{i-2}|}, \tag{11}$$

$$\mathcal{L}^{\text{nl}}_i \equiv \frac{1}{4n^3}|\mathbf{z}_{i-1} - \mathbf{z}_{i-2}|\,||\nabla\mathbf{J}(\mathbf{w}_i)||_1, \tag{12}$$

$$\mathcal{L}^{\text{acc}}_i \equiv \frac{1}{n}||U(\mathbf{w}_i) - \mathbf{M}\mathbf{w}_i||_1, \tag{13}$$

α, β, γ are tunable hyperparameters, and n is the dimensionality of the latent space. Here $\mathcal{L}^{\text{recon}}$ is the *reconstruction error*, $\mathcal{L}^{\text{pred}}$ is the *prediction error*, and both $\mathcal{L}^{\text{nl}}$ and $\mathcal{L}^{\text{acc}}$ are measures of the complexity of U. $\mathcal{L}^{\text{nl}}$ is a measure of the *nonlinearity* of the mapping U, since its Jacobian $\mathbf{J}$ will be

TABLE I. Physical systems tested.

Equations of motion	Correct	Rediscovered
Uniform motion	$\ddot{x}=0$	$\ddot{x}=0$
	$\ddot{y}=0$	$\ddot{y}=0$
Gravity	$\ddot{x}=\frac{5}{9}$	$\ddot{x}=-1$
	$\ddot{y}=-\frac{5}{9}$	$\ddot{y}=-1$
Magnetic field	$\ddot{x}=-\dot{y}$	$\ddot{x}=\frac{1}{3}\dot{y}$
	$\ddot{y}=\dot{x}$	$\ddot{y}=-\frac{1}{3}\dot{x}$
Harmonic oscillator	$\ddot{x}=-4x$	$\ddot{x}=-\frac{4}{9}x$
	$\ddot{y}=-y$	$\ddot{y}=-\frac{1}{9}y$
Quartic oscillator	$\ddot{x}=-\frac{4\times10^{-4}}{9}x(x^2+y^2)$	$\ddot{x}\approx-0.00001x(x^2+y^2)$
	$\ddot{y}=-\frac{4\times10^{-4}}{9}y(x^2+y^2)$	$\ddot{y}\approx-0.00001y(x^2+y^2)$

constant if the mapping is linear. Note that $\mathcal{L}^{\rm nl}=0$ implies that $\mathcal{L}^{\rm curv}=0$, since if $\mathbf{J}$ is constant, then $\Gamma^{\alpha}_{\mu\nu}=0$ and the curvature vanishes. Physically, $\mathcal{L}^{\rm nl}=0$ implies that the dynamics is described by coupled linear difference equations, which can be modeled by coupled linear differential equations and encompass behavior such as helical motion in magnetic fields, sinusoidal motion in harmonic oscillator potentials, and parabolic motion under gravity. $\mathcal{L}^{\rm acc}$ is a measure of the predicted *acceleration*, since there is no acceleration if the mapping is $U(\mathbf{w})=\mathbf{M}\mathbf{w}$, where

$$\mathbf{M}\equiv(-\mathbf{I}\ 2\mathbf{I}), \tag{14}$$

and $\mathbf{I}$ is the $n\times n$ identity matrix. For example, $x_i=2x_{i-1}-x_{i-2}$ gives uniform 1D motion (with i indicating the time step at which the x coordinate is recorded). An alternative implementation not requiring Jacobian gradient evaluation would be $\mathcal{L}_i^{\rm nl}\equiv\frac{1}{2n^2}||\mathbf{J}(\mathbf{w}_{i+1})-\mathbf{J}(\mathbf{w}_i)||_2^2$, and an alternative acceleration penalty would be $\mathcal{L}_i^{\rm acc}\equiv\frac{1}{n}|U(\mathbf{0})|^2+\frac{1}{2n^2}||\mathbf{M}-\mathbf{J}(\mathbf{w}_i)||_2^2$. In summary, our regularizers $\mathcal{L}^{\rm nl}$ and $\mathcal{L}^{\rm acc}$ attempt to make the time evolution as simple as possible, forcing the complexity into the autoencoder.

III. RESULTS

A. Latent space learning

We first tested our algorithm for four physical systems obeying linear differential equations, corresponding to motion with no forces, in a gravitational field, in a magnetic field, and in a 2D harmonic oscillator potential (see Table I and Fig. 4). To make things harder to solve, we defined "down" at a 45° angle for the gravity case.

For each type of motion, we generated between 100 and 150 trajectories, with around 30 video frames each, corresponding to equally spaced, consecutive time steps. For each trajectory video, the shape of the rocket and the background were kept fixed, but the position of the rocket was changed according to the corresponding physical law of motion, starting with a random initial velocity and a random initial position within the image boundaries. Our training set thus contains a total of 3000–5000 images for each type of motion; sample trajectories are shown in Fig. 4 (top), where each dot represents the x and y coordinate of the rocket in a given frame and points of the same color or shading connected by a line belong to the same trajectory. After simulating the trajectories and generating a 1000×1000 pixel image of each video frame (Fig. 1 for an example), we downsampled the image resolution to 64×64 pixels (Fig. 5) before passing them to our neural network.

The encoder network consists of five convolutional ReLU layers with kernel size 4 and padding 1, four with stride 2,

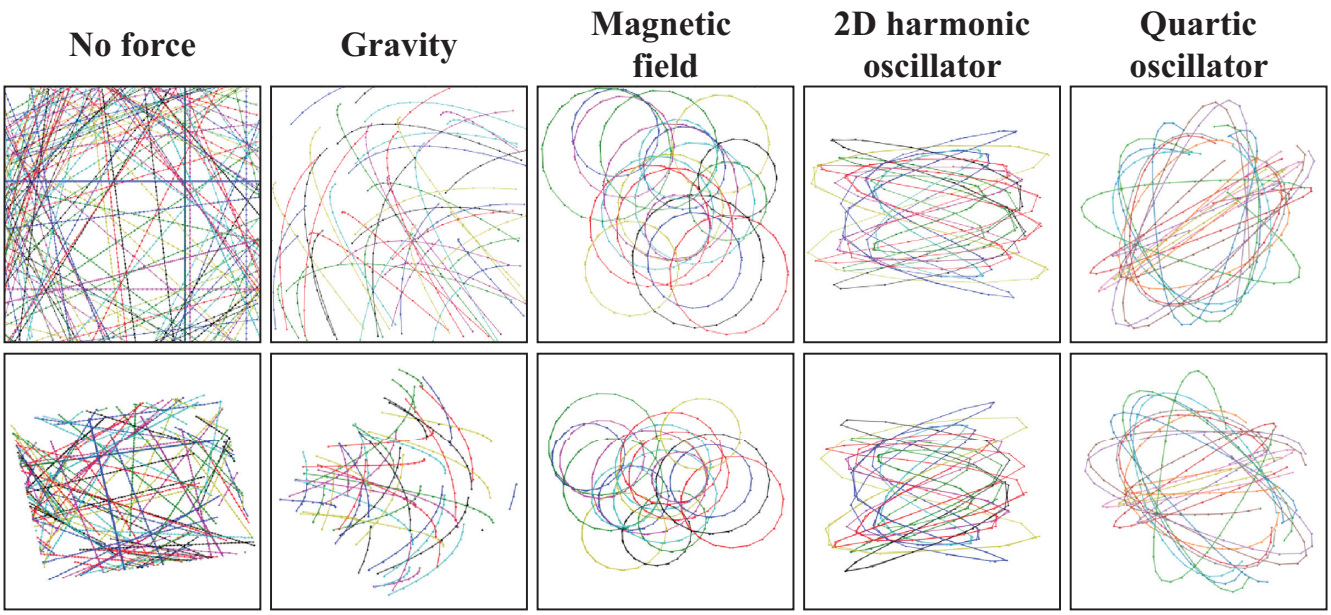

FIG. 4. Example of original (top) and rediscovered (bottom) trajectories in the latent spaces. In the top panel, each point represents the x and y coordinates of the rocket in each frame. In the bottom panel, each point corresponds to the two main principal components discovered in the 5D latent space. In both cases, points of the same color or shading and connected by a line belong to the same trajectory.

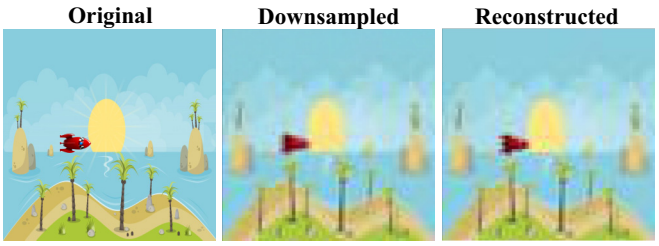

FIG. 5. Example of original, downsampled, and reconstructed image.

followed by one with stride 1. At the end there is a fully connected linear layer that reduces the output of the encoder to a vector of size equal to the dimension of the latent space. The number of channels goes from 3 for the input image to 32, 32, 64, 64, and 256 for the convolutional layers. The decoder network is a mirror image of the encoder in terms of layer dimensions, with the convolution layers replaced by deconvolution layers. The evolution operator has three fully connected 32-neuron hidden layers with softplus activation function and a linear n-neuron output layer. We implemented these networks using PyTorch using a batch size of 256 and the Adam optimizer. For these four linear types of motion, we set $\gamma = 0$ and $\alpha = \beta = 10^{-3}$ and trained for 4000 epochs with a learning rate of 10^{-3}, multiplying α and β by 10 after every 1000 epochs. We then trained for 3000 additional epochs while dividing the learning rate by 10 every 1000 epochs.

In the end, our algorithm successfully learned useful 2D latent spaces (see Fig. 4, bottom), reconstructed images with 2% rms relative error that were visually nearly indistinguishable from the truth (see Fig. 5), and achieved sub-percent prediction errors ($\mathcal{L}^{\text{pred}} \approx 0.11\%$, 0.48%, 0.31%, 0.71%, and 0.76% for the uniform motion, gravity, magnetic field, and quartic oscillator cases, respectively). However, this required overcoming two separate obstacles.

We initially lacked the factor $|\mathbf{z}_{i-1} - \mathbf{z}_{i-2}|$ in Eq. (11), so by the Alexander principle, the neural network learned to drive the prediction loss $\mathcal{L}^{\text{pred}}$ toward zero by collapsing the latent space to minuscule size. The $|\mathbf{z}_{i-1} - \mathbf{z}_{i-2}|$ factor solves this problem by making the prediction loss dimensionless and invariant under latent space rescaling.

B. Knot theory to the rescue

The second obstacle was topological. If you drop a crumpled-up towel (a 2D surface in 3D space) on the floor, it will not land perfectly flat, but with various folds. Analogously, the space of all possible rocket images forms a highly curved surface in the N-dimensional space of images, so when a randomly initialized neural network first learns to map it into a 2D latent space, there will be numerous folds. For example, the left panel of Fig. 6 shows 16 trajectories (each shown in a different color or shading) corresponding to the rocket moving uniformly in straight lines. The middle panel shows these same trajectories (with the same colors or shades as in the left panel) in the latent space first discovered by our neural network when we allowed only two latent space dimensions. Some pairs of trajectories which are supposed to be straight parallel lines (left panel) are seen to cross in a catlike pattern in the latent space (middle panel) even though they should not cross. During training, the network tries to reduce prediction and complexity loss by gradually distorting this learned latent space to give trajectories the simplest possible shapes (straight lines in this case), but gets stuck and fails to unfold the latent space. This is because the reconstruction loss $\mathcal{L}^{\text{recon}}$ effectively causes distinct images to repel each other in the latent space: if two quite different rocket images get mapped to essentially the same latent-space point, then the decoder will epically fail for at least one. Unfolding would require temporarily moving one trajectory across another, thus greatly increasing the loss. This is analogous to topological defects in physics that cannot be removed because of an insurmountable energy barrier.

Fortunately, knot theory comes to the rescue: a famous theorem states that there are no d-dimensional knots in an n-dimensional space if $n > \frac{3}{2}(1+d)$ [76]. For example, you

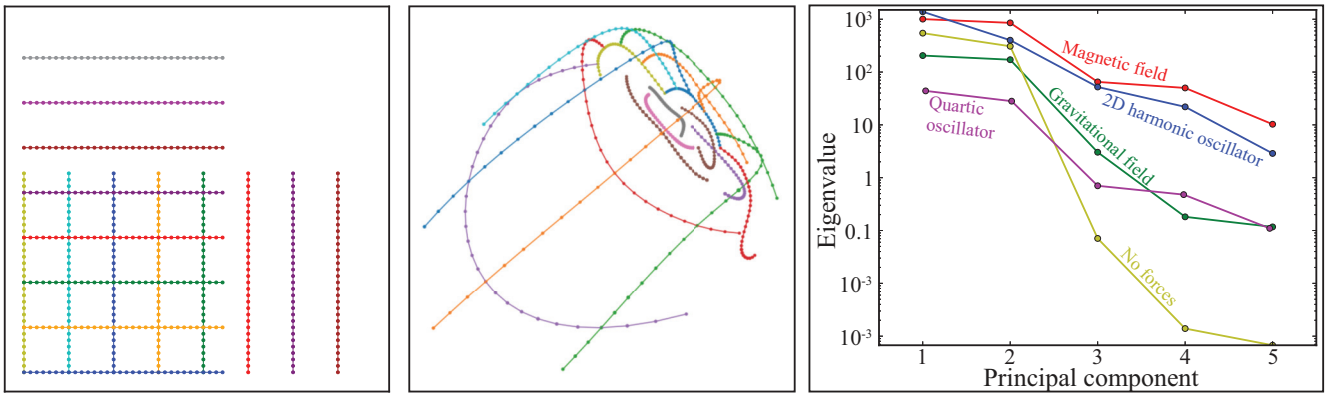

FIG. 6. The topological problems (middle) that prevented directly learning a 2-dimensional latent space (left) can be understood via knot theory and eliminated by instead discovering the two main principal components (right) in a learned 5-dimensional latent space. The left panel shows 16 force-free rocket trajectories, with points denoting the rocket center in each frame and points of the same color or shading corresponding to the same trajectory. The middle panel shows the corresponding points $\mathbf{z}_i$ produced by our encoder network trained with a 2D latent space. The right panel shows the eigenvalues obtained from a PCA on a five-dimensional learned latent space, revealing that the latent space is rather 2-dimensional because two principal components account for most of the variance.

cannot tie your shoelaces ($d = 1$) if you live in $n = 4$ dimensions. Our topological pregression problem corresponds to the inability of the neural network to untie a d-dimensional knot in n dimensions, where d is the dimensionality of the image submanifold of $\mathbb{R}^N$ ($d = 2$ for our examples). We therefore implemented the following solution, which worked well for all our examples: First run the pregression algorithm with a latent space of dimension $n' > \frac{3}{2}(1 + d)$ (we found $n' = 5$ to be enough for us) and then extract an n-dimensional latent space using principal component analysis (PCA). This corresponds to incentivizing the aforementioned towel to flatten out while still in the air and then rotating it to be parallel to the floor before landing. Figure 6 (right) shows that upon applying PCA to the points in the 5-dimensional latent space, two principal components dominate the rest (accounting for more than 90% of the variance), revealing that all rocket images get mapped roughly into a 2D plane (Fig. 4) in a 5D latent space.

To quantify the robustness of this finding, we repeated the magnetic field pregression analysis 10 times for each n' value, with random neural network initializations, obtaining success rates of 0% for $n' = 2$, 20% for $n' = 3$, 80% for $n' = 4$, and 100% for $n' = 5$. In other words, although it is possible to get lucky with smaller n', reliable success occurred only when n' exceeded the knot-theory bound $\frac{3}{2}(1 + d) = 4.5$.

C. Nonlinear dynamics and the accuracy-simplicity trade-off

Increasing the two parameters β and γ in Eq. (9) penalizes complexity ($\mathcal{L}^{\rm nl}$ and $\mathcal{L}^{\rm acc}$) more relative to inaccuracy ($\mathcal{L}^{\rm recon}$ and $\mathcal{L}^{\rm pred}$). For our quartic oscillator example (Fig. 1), achieving $\mathcal{L}^{\rm nl} = 0$ is impossible and undesirable, since the correct dynamics is nonlinear with $\nabla \mathbf{J} \neq 0$, so we wish to find the optimal tradeoff between simplicity and accuracy. We did this by training as above for 7000 epochs but setting $\beta = 0$, then keeping $\gamma = \beta$ and further training 14 networks in parallel for a geometric series of β values from 0.01 to 200. These 14 networks were trained for 3000 epochs with learning rate starting at 10^{-3} and dropping tenfold every 1000 epochs.

Since, as mentioned above, there is a broad class of equally accurate solutions related by a latent space reparametrization $\mathbf{z} \mapsto f(\mathbf{z})$, we expect that increasing β from zero to small values should discover the simplest solution in this class without decreasing prediction or reconstruction accuracy. This is the solution we want, in the spirit of Einstein's famous dictum "everything should be made as simple as possible, but not simpler." Further increasing β should simplify the solution even more, but now at the cost of leaving this equivalence class, reducing accuracy. Our numerical experiment confirmed this expectation: we could increase regularization to $\beta = 50$ (the choice shown in Fig. 1) without significant accuracy loss, after which the inaccuracy started rising abruptly. It should be noted that a similar Pareto approach could be used for the other four linear types of motions, but in those cases, the Pareto frontier would be trivial, given that the right solution (minimum loss) corresponds to having no nonlinearity (minimum complexity).

D. Image warping and noise

As mentioned in Sec. II B, the fact that our algorithm rewards simplicity in the evolution operator U rather than the encoder-decoder pair should enable it to discover the simplest possible latent space even if the space of image (x, y) coordinates is severely distorted. To test this, we replaced each image with color $c[x, y]$ (defined over the unit square) by a warped image $c'[x, y]$ defined by

$$c'[x, y] \equiv c[g(x) + x(1 - x)y, g(y) + y(1 - y)x],$$
$$g(u) \equiv u(11 - 18u + 12u^2)/5 \tag{15}$$

as illustrated in Fig. 3 (middle panel), and analyzed the 3000 warped video frames of the rocket moving in a magnetic field. As expected, the pregression algorithm recovered a non-warped latent space just as in Fig. 4, so this extra complexity was entirely absorbed by the decoder-encoder, which successfully learned the warping function $c \mapsto c'$ of Eq. (15) and its inverse.

We also tested the robustness of our pregression algorithm to noise in the form of smaller rockets added randomly to each video frame. We used 3 different types of distractor rockets as noise, and added between zero and 10 to each image as illustrated in Fig. 3 (right panel). The result was that the pregression algorithm learned to reconstruct the latent space just as before, focusing only on the large rocket, and reconstructing images with the distractor rockets removed.

E. Automatically discovering equations and inertial frames

Let us now turn to the task of discovering physical laws that are both accurate and simple. Although the five rocket-motion examples took place in the same image space, the Alexander principle implies that the five latent spaces (bottom panels in Fig. 4) will generally all be different, since we trained a separate neural network for each case. Specifically, we expect

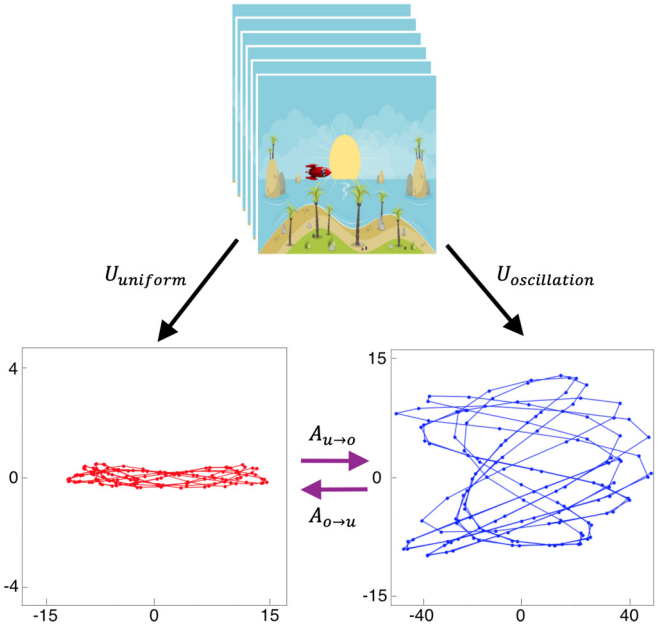

FIG. 7. Video trajectories in the 2D harmonic oscillator potential (top) look differently when mapped into the latent space using the encoder trained on that same data (right) than when mapped using the encoder trained on uniform force-free motion (left). However, the two latent spaces are equivalent up to an affine transformation.

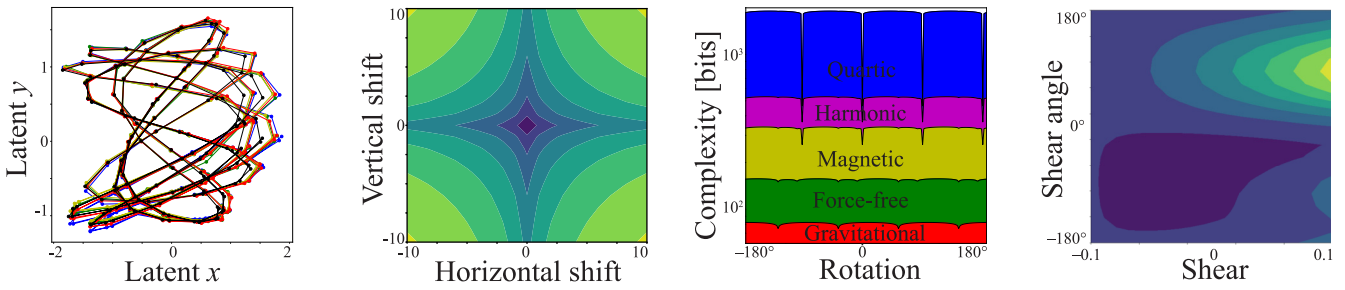

FIG. 8. Left: The five latent spaces can be unified by using affine transformations. The complexity of the equations in this unified space is minimized when having no further shift (2nd panel) or rotation (3rd panel), but a ∼3% shear (right).

the latent spaces to each differ by some affine transformation $\mathbf{r} \mapsto \mathbf{A}\mathbf{r} + \mathbf{a}$ for some constant vector $\mathbf{a}$ and 2×2 matrix $\mathbf{A}$, since affine transformations do not affect the amount of nonlinearity or acceleration required and thus leave our complexity loss functions $\mathcal{L}^{\text{acc}}$ and $\mathcal{L}^{\text{curv}}$ unchanged.

Figure 7 shows the same sequence of images of the rocket moving in the 2D harmonic oscillator potential mapped into two latent spaces, learned by training our pregression algorithm on either that same 2D oscillator data (right panel) or on the force-free data (left panel). As expected, their trajectories are seen to differ by an affine transformation. Indeed, the left panel of Fig. 8 shows that the latent spaces discovered by all our 5 experiments are interrelated by affine transformations. Here we have mapped all latent space coordinates $\mathbf{r}_i$, $i = 1, \ldots, 5$, into a single unified latent space $\mathbf{r} = \mathbf{A}_i\mathbf{r}_i + \mathbf{a}_i$ by introducing five 2×2 matrices $\mathbf{A}_i$ and 2D translation vectors $\mathbf{a}_i$ to match up corresponding rocket positions [each color is associated with an encoder trained on a specific type of motion: force free (red), 2D oscillator (blue), magnetic (yellow), gravitational field (green), and quartic oscillator (black)]. Specifically, without loss of generality, we take one of the latent spaces (derived from the harmonic oscillator) to be the unified one (so $\mathbf{r}_1 = \mathbf{r}$, $\mathbf{A}_1 = \mathbf{I}$, $\mathbf{a}_1 = 0$), and solve for the other $\mathbf{A}_i$ and $\mathbf{a}_i$ by minimizing the total mismatch

$$M \equiv \sum_{i=2}^{5} \left\langle \ell(|\mathbf{A}_i\mathbf{r}_i + \mathbf{a}_i - \mathbf{r}_1|) + \ell\left(\left|\mathbf{A}_i^{-1}(\mathbf{r}_1 - \mathbf{a}_i) - \mathbf{r}_i\right|\right)\right\rangle, \tag{16}$$

where the average is over all our rocket images mapped through the the five encoders. If the loss function penalizing mismatch distance were $\ell(r) = r^2$, Eq. (16) would simply be a χ^2 minimization determining $\mathbf{A}_i$ and $\mathbf{a}_i$ via linear regression, except that we have also penalized inaccuracy in the inverse mapping (second term) to avoid biasing $\mathbf{A}_i$ low. To increase robustness toward outliers, we instead followed the prescription of [26] by choosing $\ell(r) \equiv \frac{1}{2}\log_2[1 + (r/\epsilon)^2]$ and minimizing M with gradient descent, using an annealing schedule $\epsilon = 10^1, 10^0, ..., 10^{-10}$.

Next, we estimated the velocity $\dot{\mathbf{r}}$ and acceleration $\ddot{\mathbf{r}}$ at each data point by cubic spline fitting to each trajectory $\mathbf{r}(t)$ in the unified latent space, and found candidate differential equations of the form $\ddot{\mathbf{r}} = f(\dot{\mathbf{r}}, \mathbf{r})$ using the publicly available AI Feynman symbolic regression package [22,23]. To eliminate dependence on the cubic spline approximation, we then recomputed the accuracy of each candidate formula f by using it to predict each data point from its two predecessors using the boundary-value ODE solver "scipy.integrate.solve_bvp" [77], selecting the most accurate formula for each of our five examples.

Applying an affine transformation $\mathbf{r} \mapsto \mathbf{A}\mathbf{r} + \mathbf{a}$ to both the data and these equations of course leaves the prediction accuracy unchanged, so we now exploit this to further reduce the total information-theoretic complexity of our equations, defined as in [23,26] and summarized in Table II. Figure 8 (2nd panel) shows a contour plot of the equations' complexity as a function of an overall shift (darker means smaller). We observe a clear optimum for the shift vector $\mathbf{a}$, corresponding to eliminating additive constants in the harmonic and quartic oscillator equations. For example, $\ddot{x} = -x$ is simpler than $\ddot{x} = 2.236 - x$. The 3rd panel of Fig. 8 shows the total equation complexity (as a stacked histogram) as a function of an overall rotation of the coordinate axes. We see several minima: the gravitational example likes 45° rotation because this makes the new horizontal acceleration vanish, but the other examples outvote it in favor of a 0° rotation to avoid xy cross terms.

Only three degrees of freedom now remain in our matrix $\mathbf{A}$: two for shear (expanding along some axis and shrinking by the inverse factor along the perpendicular axis) and one for an overall scaling. We apply the "vectorSnap" algorithm of [23] to reveal rational ratios between parameters and then select the shear that maximizes total accuracy. As can be seen in the contour plot in the right panel of Fig. 8 (darker color means lower complexity), ∼3% shear is optimal. Finally, we apply the overall scaling that minimizes total complexity, resulting in the rediscovered laws of motion shown in the right column of Table I. The table shows that these are in fact exactly the laws of motion used to generate our training set images (up to some noise in the quartic term prefactor caused by the cubic spline interpolation), but reexpressed in a latent space that is larger by a factor $\frac{9}{5}$ than the one we used, and has its x axis flipped, which further simplifies our formulas (for example, the rediscovered gravitational acceleration is 1 instead of $\frac{5}{9}$).

We note that the formulas in Table I were discovered fully automatically in the sense that no hyperparameters were adjusted in any of the steps (except for the latent space dimensionality being reset to exceed the knot-theory requirement as mentioned above). These laws of motion were autodiscovered in three conceptually different steps:

TABLE II. Complexity definitions.

Object	Symbol	Description length DL
Natural number	n	$\log_2 n$
Integer	m	$\log_2(1+\lvert m\rvert)$
Rational number	m/n	$\mathrm{DL}(m)+\mathrm{DL}(n)=\log_2[(1+\lvert m\rvert)n]$
Real number	r	$\log_+(\frac{r}{\varepsilon}),\quad \log_+(x)\equiv\frac{1}{2}\log_2(1+x^2)$
Parameter vector	$\mathbf{p}$	$\sum_i \mathrm{DL}(p_i)$
Parametrized function	$f(\mathbf{x};\mathbf{p})$	$\mathrm{DL}(\mathbf{p})+k\log_2 n$; n basis functions appear k times

(1) Train a neural network to find an accurate black-box fit. There is a continuum of equally accurate fits corresponding to latent-space reparametrization invariance.

(2) Exploit this reparametrization invariance to minimize complexity caused by unnecessary nonlinearity. There remains a continuum of equally accurate fits corresponding to latent-space affine transformations.

(3) Exploit this affine invariance to minimize total symbolic complexity in the fitted equations.

Steps 2 and 3 both reveal dynamical simplicity by absorbing ever more of the complexity into the autoencoder.

IV. SUMMARY

We have presented a method for unsupervised learning of equations of motion for objects in raw and optionally distorted unlabeled video. This automatic undistortion may be helpful for modeling real-world video afflicted by stereoscopic projection, lens artifacts, varying lighting conditions, etc., and also for learning degrees of freedom such as 3D coordinates and rotation angles. Our method is in no way limited to video, and can be applied to any time-evolving data set, say N numbers measured by a set of sensors, where one is interested in discovering predictable features in this time-series data and learning their laws of motion in as simple form as possible.

Although we focused on dynamics, it could also be interesting to generalize our approach to other situations, by attempting to infer other properties of the system rather than its future state. Another interesting avenue for future work is to explore whether the above-mentioned topological intuition provided by knot theory can help improve autoencoders more generally, and whether the above-mentioned regularization of curvature or nonlinearity can prove useful in other machine-learning contexts.

The reparametrization invariance of general relativity teaches us that there is an infinite class of coordinate systems that provide equally valid physical descriptions. In our case, such reparametrization invariance of our autodiscovered latent space is a nuisance because, in contrast to general relativity, the laws of motion are not reparametrization invariant and can be made arbitrarily complicated. We broke this degeneracy by quantifying and minimizing the geometric and symbolic complexity of the dynamics. Although different systems were simplest in different coordinate systems, we found that minimizing total complexity for all of them recovered a standard isotropic inertial frame. An interesting topic for future work would be to explore whether our brains' representations of physical systems are similarly optimized to make prediction as simple as possible.

ACKNOWLEDGMENTS

The authors with to thank Zhiyu Dong, Jiahai Feng, Bhairav Mehta, Andrew Tan, and Tailin Wu for helpful comments, and the Center for Brains, Minds, and Machines (CBMM) for hospitality. This work was supported by the Institute for AI and Fundamental Interactions through NSF Grant No. PHY-2019786, the Casey and Family Foundation, the Ethics and Governance of AI Fund, the Foundational Questions Institute, the Rothberg Family Fund for Cognitive Science, the Templeton World Charity Foundation, Inc., and the Office of Nuclear Physics of the US Department of Energy under Grant No. DE-SC0021179.

[1] A. Koyré, *The Astronomical Revolution: Copernicus-Kepler-Borelli* (Routledge, London, United Kingdom, 2013).

[2] J. P. Crutchfield and B. S. McNamara, Equation of motion from a data series, Complex Syst. **1**, 121 (1987).

[3] S. Dzeroski and L. Todorovski, Discovering dynamics: From inductive logic programming to machine discovery, J. Intell. Inf. Syst. **4**, 89 (1995).

[4] E. Bradley, M. Easley, and R. Stolle, Reasoning about nonlinear system identification, Artif. Intell. **133**, 139 (2001).

[5] P. Langley, D. George, S. D. Bay, and K. Saito, Robust induction of process models from time-series data, in *Proceedings of the 20th International Conference on Machine Learning (ICML-03)* (AAAI Press, Palo Alto, California, 2003), pp. 432–439.

[6] M. Schmidt and H. Lipson, Distilling free-form natural laws from experimental data, Science **324**, 81 (2009).

[7] R. K. McRee, Symbolic regression using nearest neighbor indexing, in *Proceedings of the 12th Annual Conference Companion on Genetic and Evolutionary Computation* (ACM, NY, USA, 2010), pp. 1983–1990.

[8] D. P. Searson, D. E. Leahy, and M. J. Willis, GPTIPS: An open source genetic programming toolbox for multigene symbolic regression, in *Proceedings of the International Multiconference of Engineers and Computer Scientists*, Vol. 1 (IMECS, Hong Kong, 2010), pp. 77–80.

[9] R. Dubčáková, Eureqa: Software review, Genet. Program. Evolvable Mach. **12**, 173 (2011).

[10] S. Stijven, W. Minnebo, and K. Vladislavleva, Separating the wheat from the chaff: On feature selection and feature importance in regression random forests and symbolic regression, in *Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation* (ACM, New York, USA, 2011), pp. 623–630.

[11] M. D. Schmidt, R. R. Vallabhajosyula, J. W. Jenkins, J. E. Hood, A. S. Soni, J. P. Wikswo, and H. Lipson, Automated refinement and inference of analytical models for metabolic networks, Phys. Biol. **8**, 055011 (2011).

[12] C. Hillar and F. Sommer, Comment on the article "Distilling free-form natural laws from experimental data", arXiv:1210.7273.

[13] B. C. Daniels and I. Nemenman, Automated adaptive inference of phenomenological dynamical models, Nat. Commun. **6**, 8133 (2015).

[14] P. Langley and A. Arvay, Heuristic induction of rate-based process models, in *Twenty-Ninth AAAI Conference on Artificial Intelligence* (AAAI, Palo Alto, California, 2015).

[15] I. Arnaldo, U.-M. O'Reilly, and K. Veeramachaneni, Building predictive models via feature synthesis, in *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation* (ACM, New York, USA, 2015), pp. 983–990.

[16] S. L. Brunton, J. L. Proctor, and J. N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, Proc. Natl. Acad. Sci. USA **113**, 3932 (2016).

[17] M. Guzdial, B. Li, and M. O. Riedl, Game engine learning from video, in *2017 International Joint Conference on Artificial Intelligence* (IJCAI, 2017), pp. 3707–3713.

[18] M. Quade, M. Abel, J. N. Kutz, and S. L. Brunton, Sparse identification of nonlinear dynamics for rapid model recovery, Chaos **28**, 063116 (2018).

[19] M. Koch-Janusz and Z. Ringel, Mutual information, neural networks, and the renormalization group, Nat. Phys. **14**, 578 (2018).

[20] W. Kong, C. Liaw, A. Mehta, and D. Sivakumar, A new dog learns old tricks: RL finds classic optimization algorithms, in *International Conference on Learning Representations* (ICLR, 2019).

[21] J. Liang and X. Zhu, Phillips-inspired machine learning for band gap and exciton binding energy prediction, J. Phys. Chem. Lett. **10**, 5640 (2019).

[22] S.-M. Udrescu and M. Tegmark, AI Feynman: A physics-inspired method for symbolic regression, Sci. Adv. **6**, eaay2631 (2020).

[23] S.-M. Udrescu, A. Tan, J. Feng, O. Neto, T. Wu, and M. Tegmark, AI feynman 2.0: Pareto-optimal symbolic regression exploiting graph modularity, in *Advances in Neural Information Processing Systems*, edited by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Curran Associates, Inc., 2020), Vol. 33, pp. 4860–4871.

[24] E. P. Wigner, Invariance in physical theory, Proc. Am. Philos. Soc. **93**, 521 (1949).

[25] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, Nature (London) **521**, 436 (2015).

[26] T. Wu and M. Tegmark, Toward an artificial intelligence physicist for unsupervised learning, Phys. Rev. E **100**, 033311 (2019).

[27] R. Iten, T. Metger, H. Wilming, L. Del Rio, and R. Renner, Discovering Physical Concepts with Neural Networks, Phys. Rev. Lett. **124**, 010508 (2020).

[28] M. A. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, and S. Chopra, Video (language) modeling: A baseline for generative models of natural videos, arXiv:1412.6604.

[29] V. Michalski, R. Memisevic, and K. Konda, Modeling deep temporal dependencies with recurrent grammar cells, in *Advances in Neural Information Processing Systems* (ACM, New York, USA, 2014), pp. 1925–1933.

[30] N. Srivastava, E. Mansimov, and R. Salakhudinov, Unsupervised learning of video representations using LSTMs, in *International Conference on Machine Learning* (AAAI Press, Palo Alto, California, 2015), pp. 843–852.

[31] J. Oh, X. Guo, H. Lee, R. L. Lewis, and S. Singh, Action-conditional video prediction using deep networks in Atari games, in *Advances in Neural Information Processing Systems* (ACM, New York, USA, 2015), pp. 2863–2871.

[32] C. Finn, I. Goodfellow, and S. Levine, Unsupervised learning for physical interaction through video prediction, in *Advances in Neural Information Processing Systems* (ACM, New York, USA, 2016), pp. 64–72.

[33] W. Lotter, G. Kreiman, and D. Cox, Deep predictive coding networks for video prediction and unsupervised learning, arXiv:1605.08104.

[34] F. Cricri, X. Ni, M. Honkala, E. Aksu, and M. Gabbouj, Video ladder networks, arXiv:1612.01756.

[35] N. Kalchbrenner, A. van den Oord, K. Simonyan, I. Danihelka, O. Vinyals, A. Graves, and K. Kavukcuoglu, Video pixel networks, in *Proceedings of the 34th International Conference on Machine Learning*, Vol. 70 (JMLR, 2017), pp. 1771–1779.

[36] X. Liang, L. Lee, W. Dai, and E. P. Xing, Dual motion GAN for future-flow embedded video prediction, in *Proceedings of the IEEE International Conference on Computer Vision* (IEEE, Piscataway, 2017), pp. 1744–1752.

[37] E. L. Denton *et al.*, Unsupervised learning of disentangled representations from video, in *Advances in Neural Information Processing Systems* (ACM, New York, USA, 2017), pp. 4414–4423.

[38] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee, Decomposing motion and content for natural video sequence prediction, arXiv:1706.08033.

[39] V. Vukotić, S.-L. Pintea, C. Raymond, G. Gravier, and J. C. Van Gemert, One-step time-dependent future video frame prediction with a convolutional encoder-decoder neural network, in *International Conference on Image Analysis and Processing* (Springer, Berlin, Germany, 2017), pp. 140–151.

[40] M. Babaeizadeh, C. Finn, D. Erhan, R. H. Campbell, and S. Levine, Stochastic variational video prediction, arXiv:1710.11252.

[41] M. Oliu, J. Selva, and S. Escalera, Folded recurrent neural networks for future video prediction, in *Proceedings of the European Conference on Computer Vision (ECCV)* (CVF, Berlin, Germany, 2018), pp. 716–731.

[42] Z. Liu, X. Chai, and X. Chen, Deep memory and prediction neural network for video prediction, Neurocomputing **331**, 235 (2019).

[43] J. Carrasquilla and R. G. Melko, Machine learning phases of matter, Nat. Phys. **13**, 431 (2017).

[44] E. P. L. Van Nieuwenburg, Y.-H. Liu, and S. D. Huber, Learning phase transitions by confusion, Nat. Phys. **13**, 435 (2017).
[45] E. van Nieuwenburg, E. Bairey, and G. Refael, Learning phase transitions from dynamics, Phys. Rev. B **98**, 060301(R) (2018).
[46] G. Torlai and R. G. Melko, Learning thermodynamics with Boltzmann machines, Phys. Rev. B **94**, 165134 (2016).
[47] T. Ohtsuki and T. Ohtsuki, Deep learning the quantum phase transitions in random electron systems: Applications to three dimensions, J. Phys. Soc. Jpn. **86**, 044708 (2017).
[48] V. Dunjko and H. J. Briegel, Machine learning and artificial intelligence in the quantum domain: A review of recent progress, Rep. Prog. Phys. **81**, 074001 (2018).
[49] I. Yildirim, K. A. Smith, M. Belledonne, J. Wu, and J. B. Tenenbaum, Neurocomputational modeling of human physical scene understanding, in *2nd Conference on Cognitive Computational Neuroscience* (CCN, 2018).
[50] D. Zheng, V. Luo, J. Wu, and J. B. Tenenbaum, Unsupervised learning of latent physical properties using perception-prediction networks, arXiv:1807.09244.
[51] M. B. Chang, T. Ullman, A. Torralba, and J. B. Tenenbaum, A compositional object-based approach to learning physical dynamics, arXiv:1612.00341.
[52] Z. Zhang, Y. Zhao, J. Liu, S. Wang, R. Tao, R. Xin, and J. Zhang, A general deep learning framework for network reconstruction and dynamics learning, Appl. Network Sci. **4**, 1 (2019).
[53] S. Russell, D. Dewey, and M. Tegmark, Research priorities for robust and beneficial artificial intelligence, AI Mag. **36**, 105 (2015).
[54] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, Concrete problems in AI safety, arXiv:1606.06565.
[55] M. Boden, J. Bryson, D. Caldwell, K. Dautenhahn, L. Edwards, S. Kember, P. Newman, V. Parry, G. Pegman, T. Rodden *et al.*, Principles of robotics: Regulating robots in the real world, Connect. Sci. **29**, 124 (2017).
[56] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner *et al.*, Relational inductive biases, deep learning, and graph networks, arXiv:1806.01261.
[57] K. S. Bhat, S. M. Seitz, J. Popović, and P. K. Khosla, Computing the physical parameters of rigid-body motion from video, in *European Conference on Computer Vision* (Springer, Berlin, Germany, 2002), pp. 551–565.
[58] J. Tompson, K. Schlachter, P. Sprechmann, and K. Perlin, Accelerating Eulerian fluid simulation with convolutional networks, in *Proceedings of the 34th International Conference on Machine Learning*, Vol. 70 (JMLR, 2017), pp. 3424–3433.
[59] P. Y. Lu, S. Kim, and M. Soljačić, Extracting Interpretable physical parameters from spatiotemporal systems using unsupervised learning, Phys. Rev. X **10**, 031056 (2020).
[60] J. Wu, J. J. Lim, H. Zhang, J. B. Tenenbaum, and W. T. Freeman, Physics 101: Learning physical object properties from unlabeled videos, in *British Machine Vision Conference*, Vol. 2 (BMVC, 2016), p. 7.
[61] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap, A simple neural network module for relational reasoning, in *Advances in Neural Information Processing Systems* (ACM, New York, USA, 2017), pp. 4967–4976.
[62] R. Liu, J. Lehman, P. Molino, F. P. Such, E. Frank, A. Sergeev, and J. Yosinski, An intriguing failing of convolutional neural networks and the CoordConv solution, in *Advances in Neural Information Processing Systems* (ACM, New York, USA, 2018), pp. 9605–9616.
[63] J. B. Hamrick, K. R. Allen, V. Bapst, T. Zhu, K. R. McKee, J. B. Tenenbaum, and P. W. Battaglia, Relational inductive bias for physical construction in humans and machines, arXiv:1806.01203.
[64] N. Watters, D. Zoran, T. Weber, P. Battaglia, R. Pascanu, and A. Tacchetti, Visual interaction networks: Learning a physics simulator from video, in *Advances in Neural Information Processing Systems* (ACM, New York, USA, 2017), pp. 4539–4547.
[65] A. van den Oord, Y. Li, and O. Vinyals, Representation learning with contrastive predictive coding, arXiv:1807.03748.
[66] S. Greydanus, M. Dzamba, and J. Yosinski, Hamiltonian neural networks, in *Advances in Neural Information Processing Systems* (ACM, New York, USA, 2019), pp. 15353–15363.
[67] H. Bourlard and Y. Kamp, Auto-association by multilayer perceptrons and singular value decomposition, Biol. Cybern. **59**, 291 (1988).
[68] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, Backpropagation applied to handwritten zip code recognition, Neural Comput. **1**, 541 (1989).
[69] G. E. Hinton and R. S. Zemel, Autoencoders, minimum description length, and Helmholtz free energy, in *Advances in Neural Information Processing Systems* (ACM, New York, USA, 1994), pp. 3–10.
[70] G. E. Hinton and R. R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science **313**, 504 (2006).
[71] Y. Bengio, A. C. Courville, and P. Vincent, Unsupervised feature learning and deep learning: A review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. **35**, 1798 (2013).
[72] D. P. Kingma and M. Welling, Auto-encoding variational Bayes, arXiv:1312.6114.
[73] Y. Bengio, Deep learning of representations: Looking forward, in *International Conference on Statistical Language and Speech Processing* (Springer, Berlin, Germany, 2013), pp. 1–37.
[74] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, beta-VAE: Learning basic visual concepts with a constrained variational framework, in *International Conference on Learning Representations*, Vol. 2 (ICLR, 2017), p. 6.
[75] A. Achille and S. Soatto, Information dropout: Learning optimal representations through noisy computation, IEEE Trans. Pattern Anal. Mach. Intell. **40**, 2897 (2018).
[76] E. C. Zeeman, Unknotting spheres, Ann. Math. **72**, 350 (1960).
[77] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, SciPy 1.0: Fundamental algorithms for scientific computing in Python, Nat. Methods **17**, 261 (2020).