# Learning Physics From Video: Unsupervised Physical Parameter Estimation for Continuous Dynamical Systems

Alejandro Castañeda Garcia  Jan Warchocki  Jan van Gemert  Daan Brinks  Nergis Tömen
Delft University of Technology

## Abstract

*Extracting physical dynamical system parameters from recorded observations is key in natural science. Current methods for automatic parameter estimation from video train supervised deep networks on large datasets. Such datasets require labels, which are difficult to acquire. While some unsupervised techniques–which depend on frame prediction–exist, they suffer from long training times, initialization instabilities, only consider motion-based dynamical systems, and are evaluated mainly on synthetic data. In this work, we propose an unsupervised method to estimate the physical parameters of known, continuous governing equations from single videos suitable for different dynamical systems beyond motion and robust to initialization. Moreover, we remove the need for frame prediction by implementing a KL-divergence-based loss function in the latent space, which avoids convergence to trivial solutions and reduces model size and compute. We first evaluate our model on synthetic data, as commonly done. After which, we take the field closer to reality by recording Delfys75: our own real-world dataset of 75 videos for five different types of dynamical systems to evaluate our method and others. Our method compares favorably to others. Code and data are available online:* https://github.com/Alejandro-neuro/Learning_physics_from_video.

## 1. Introduction

Estimating dynamical parameters of physical and biological systems from videos allows relating visual data to known governing equations which can be used to make predictions, improve mathematical models, understand diseases, and, in general, advance our knowledge in science and technology [7, 20, 37]. Use cases include trajectory prediction for celestial objects [17], healthy and diseased tissue characterization [16], and physical model validation [7, 8].

Fitting governing equations is an inverse problem [2], which often requires using additional sensors to directly measure system states. Video-based measurements can eliminate the need for additional sensors, yet, require manually labelling pixels or video frames which is time-consuming and expensive. Therefore, automated and unsupervised methods are key to extract dynamics from videos and accurately estimate physical parameters [7, 17, 19, 20, 37].

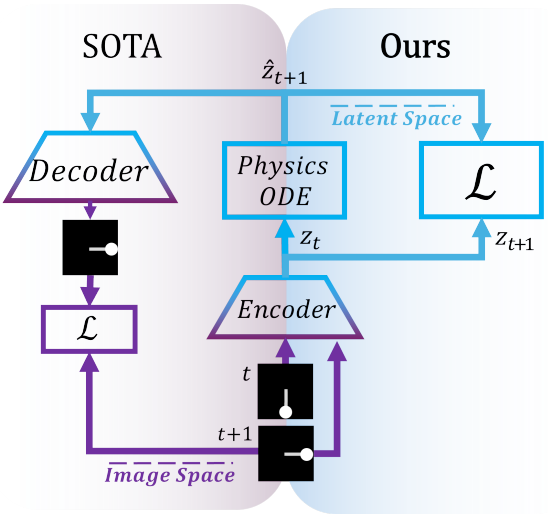



Figure 1. We propose a novel unsupervised approach to physical parameter estimation from videos. Black squares are video frames with different states of a white pendulum. Starting from a frame at time $t$ (center) an encoder estimates the dynamical states $z_t$. A learnable physics block (Physics ODE) solves the dynamical system equations to predict future states $\hat{z}_{t+1}$ in latent space (blue lines). Previous state-of-the-art methods (left) then use decoders and a reconstruction loss ($\mathcal{L}$, purple left) to train the physics ODE block. In contrast, our method (right) completely avoids the need for a decoder by leveraging a loss function in the latent space ($\mathcal{L}$, blue right). Our loss function minimizes the distance between the estimated states $\hat{z}_{t+1}$ and $z_{t+1}$.

Recent work addressed parameter estimation from video by deep learning [7, 8, 43] or reinforcement learning [3]. Supervised methods rely on datasets with extensive and high precision labels which are exceedingly difficult to obtain [1, 4, 29, 31, 40]. To avoid labeling, current unsupervised methods for estimating physical parameters build on encoder-decoder network designs: reconstructing video frames from low-dimensional representations. However,

frame reconstruction is a mere by-product of the parameter estimation and leads to hardwired architectures which cannot be extended to different dynamics [17, 20]. Consequently, current solutions [17, 20, 23, 44, 51] are constrained to motion dynamics, excluding a wide variety of systems with dynamics related to brightness, color, and deformations, among others [17, 20]. While such methods have been shown to work well on synthetic data [17, 20, 49], they are resource intensive, sensitive to initialization, and it is not clear how well they will perform on realistic data.

Here, we propose an unsupervised method to solve the inverse problem using videos of dynamical systems governed by known continuous equations. Unlike prior approaches, our method is versatile across various systems beyond motion. We use a loss in latent space that eliminates the need for a reconstruction decoder. This approach is faster, less resource-intensive, and more robust to initial conditions than existing methods. Additionally, to take a step towards real-world applications, we collect the Delfys75 dataset: 75 real-world videos across five dynamical systems, with ground-truth parameters for motion, brightness, and scale dynamics. We benchmark baselines and our model, where we compare favorably. The proposed method is visualised in Figure 1. Our main contributions are summarized as:

- Accurate dynamical state estimation from video with precise extrapolation at test time.
- A decoder-free model, capable of modeling dynamics beyond motion, robust to changes in initial conditions.
- An unsupervised latent space loss which avoids model collapse in unsupervised parameter estimation.
- Delfys75: a real-world dataset of 75 videos for 5 different physical systems with annotated ground truth.

## 2. Related Work

**Physics and deep learning.** The relationship between physics and deep learning is symbiotic: Physics inspired segmentation [28] and generative models [38, 39] as well as the design of new architectures [14]. Likewise, deep learning is used to study, understand and create new physics from data [7, 8, 19, 43]. Techniques like physics-informed neural networks (PINN) [24, 35] or Lagrangian neural networks (LLN) [9, 30] are designed to solve inverse problems. Yet, PINNs are usually constrained to initial conditions, boundary conditions, and time reference [13, 24, 32]. Moreover, these methods are supervised and require labeled data. Because obtaining labeled data in inverse problems is expensive or even infeasible [1] we avoid the need for labels by proposing an unsupervised method, following [1, 4, 29, 31, 40].

While some methods incorporate physics knowledge or design inspiration, they focus on future predictions rather than estimating physical parameters [5, 21, 34, 42]. Alternatively, some approaches attempt to relearn or propose new equations [8, 12, 27]. Our method differs by using known governing equations to both predict on latent space and learn physical parameters, which prevents direct comparisons with these techniques. Moreover, some of these approaches not suitable for video applications.

**Learning physics from video.** Research on learning physics from videos often focuses on frame prediction [6, 11, 15, 26] and not on accurate parameter estimation. Existing works on extracting physical information from videos [47] consider physical parameters (e.g. the friction coefficient) or the value of the dynamical state variable (e.g. the position or velocity), but employ supervised methods which require labelled datasets with access to the dynamical variables' or parameters' ground truth [10, 30, 35, 43, 45, 47, 49, 50]. Moreover, these methods mainly address motion dynamics and are based on systems similar to interaction networks [6, 41, 43]. Some methods [22, 44] aim to parameterize and solve differential equations to simulate the deformations. However, dynamical systems are not limited to motion, and we propose a method that goes beyond motion, illustrated with use-cases for intensity changes and scaling.

**Unsupervised parameter estimation from video.** Existing unsupervised work uses frame representations and physics priors of known governing equations with unknown parameters to estimate [17, 20, 23, 44, 51]. Some models [23, 51] use variational auto-encoders (VAE) [25] with a physics engine between the encoder and the decoder to estimate parameters; however, the reconstructions are poor, constraining the model to simple motion problems. Approaches similar to ours [17, 20] estimate parameters from a single video without annotations, and constitute our baselines. Empirically, our method compares favorably in terms of robustness to initializations, analysis of latent space dynamics and extension to different systems.

**Datasets.** One dataset for unsupervised parameter estimation from video is Physics101 [46], which includes recordings of physical experiments, but lacks parameter ground truths and object masks required by some models [17]. Consequently, Physics101 has not been used for unsupervised parameter estimation, and existing models are primarily tested on synthetic datasets, with minimal overlap across studies [17, 20, 49]. In this paper, we introduce Delfys75: a new dataset featuring real-world videos of diverse experiments, estimated physical parameters, and object masks.

**Baselines.** As a strong unsupervised parameter estimator, Jaques et al. [20] uses a traditional auto-encoder with a physics engine in the latent space to reconstruct inputs and generate future frame predictions. The model also incorporates a U-Net in the encoder to learn segmentation masks for the object of interest in an unsupervised way; the capability to learn this mask is linked to the spatial transformer used in the decoder similar to [18]. During inference, the spatial transformer is used to perform affine transformations on the

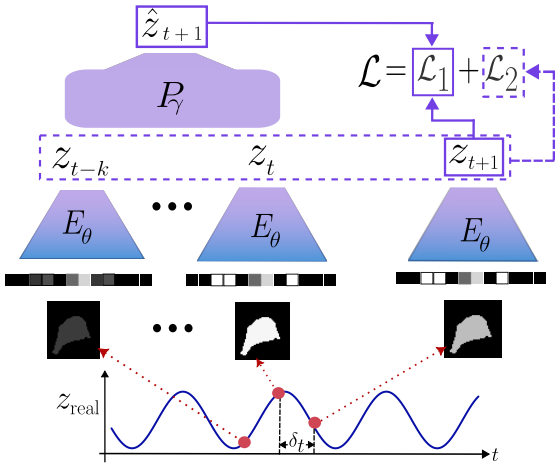



Figure 2. Method overview. A video recording of an object with a periodic brightness change (bottom) displays dynamics $z_{\text{real}}$ with sampling period $\delta t$. Each frame is mapped by the encoder $E_\theta$ to the unsupervised latent representation $z_t$. The physics block $P_\gamma$ generates a prediction of the future step $\hat{z}_{t+1}$, which we compare to the encoded representation $z_{t+1}$ of frame $t+1$. **Top-right:** Loss function of our model; the first term ensures the prediction fits with the encoding, while the second expression controls the variance of $z$. This image summarizes our methodology and the relationship between the different blocks.

mask to 'move' the object in future frame predictions. This limits the application of [20] to motion dynamics.

On the other hand, Hofherr et al. [17] uses a differentiable ODE solver to estimate the parameters. This model also uses a spatial transformer, but at the pixel level: Object pixels are displaced using predictions made by the ODE solver. The model needs to be trained with the use of masks, to learn which pixels are displaced in future frames.

Taken together, reconstructing frames using only low dimensional data (e.g. a set of positions and velocities) is challenging and slows down training for parameter estimation. Therefore, [17, 20, 44] had to limit their scope to using a mask and a spatial transformer [18], excluding dynamical systems with changes in intensity and colour, deformations and non-uniform scaling among others which we explicitly allow in our paper.

## 3. Model

Our model estimates the parameters of a known governing equation from a video recording. Since it is unsupervised, the training set consists of unannotated frames with a known frame rate denoted as $\delta t$. We do not reconstruct frames and thus have only a simple encoder and a physics block. In Figure 2 we show our approach.

**Scope.** We study systems represented by autonomous differential equations (Eq. 1), which depend only on the state variable captured in video. Assuming no external forces,

$$z^{(n)} + \gamma_n z^{(n-1)} + \ldots + \gamma_2 z^{(1)} + \gamma_1 z + \gamma_0 = 0 | z^{(k)} = \frac{d^k z}{dt^k} \tag{1}$$

is an $n^{\text{th}}$-order system, where $z$ is the time-dependent state variable or "dynamic variable", $z^{(k)}$ with $k = 1, 2, \ldots n$ is the $k^{\text{th}}$-derivative of $z$ with respect to time $t$ and $\gamma_i$ with $i = 0, 1, \ldots n-1$ are the parameters of the equation we want to estimate.

While our approach can be extended to equations of arbitrary order, in the following *proof-of-concept*, we first consider a second-order differential equation since it is the maximum order used in previous work:

$$z^{(2)} + \gamma_1 z^{(1)} + \gamma_0 z = 0. \tag{2}$$

**Physics block.** Our physics block numerically solves the differential equation using a single step of Euler's method:

$$z_t^{(1)} \approx \tfrac{z_{t+1} - z_t}{\delta t} \approx \tfrac{z_t - z_{t-1}}{\delta t} \tag{3}$$

$$z_{t+1} = z_t + \delta t z_t^{(1)} \tag{4}$$

$$z_{t+1}^{(1)} = z_t^{(1)} + \delta t z_t^{(2)}. \tag{5}$$

Plugging in Eq. 2, we can rewrite the physics block as:

$$\hat{z}_{t+1} = z_t + \delta t \left( z_t^{(1)} - \delta t (\gamma_1 z_t^{(1)} + \gamma_0 z_t) \right). \tag{6}$$

$$P : \mathbb{R}^d \to \mathbb{R}^d,$$

$$\hat{z}_{t+1} = P_\gamma(z_t, .. z_{t-n}; \gamma). \tag{7}$$

where $\gamma_i$ are learnable parameters and the predicted latent space $\hat{z}$ for time $t+1$ is a function $P_\gamma(\cdot)$ of the latent representations of the $n$ previous frames. We use the notation $\hat{\mathbf{z}}$ for the latent space predicted using the function $P$, which is different from $\mathbf{z}$ predicted by the encoder.

**Encoder.** The encoder is a network $E_\theta(x)$ that maps images $x \in \mathbb{R}^{w \times h \times c}$ to the state variable $z \in \mathbb{R}^d$, where $d$ is the number of dynamic variables. We use an MLP, similar to [20] suitable as localization networks, with three layers and the ReLU activation function for the mapping

$$z_t = E_\theta(x_t), \quad \text{where} \quad E : \mathbb{R}^{w \times h \times c} \to \mathbb{R}^d. \tag{8}$$

**Predictions.** The encoder maps images $x_t \in \mathbb{R}^{w \times h \times c}$ to the state variable $z_t \in \mathbb{R}^d$ for all time steps $t \in [0, T]$ leading to $\hat{z}$ with dimensionality $\mathbb{R}^{T \times d}$ for all input frames. In particular, we have the following two vectors:

$$\mathbf{z} = \begin{bmatrix} z_n \\ \vdots \\ z_{t+1} \\ \vdots \\ z_T \end{bmatrix} = \begin{bmatrix} E_\theta(x_n) \\ \vdots \\ E_\theta(x_{t+1}) \\ \vdots \\ E_\theta(z_T) \end{bmatrix} \tag{9}$$

$$\hat{\mathbf{z}} = \begin{bmatrix} \hat{z}_n \\ \vdots \\ \hat{z}_{t+1} \\ \vdots \\ \hat{z}_T \end{bmatrix} = \begin{bmatrix} P_\gamma(z_{n-1}, \dots, z_0; \gamma) \\ \vdots \\ P_\gamma(z_t, \dots, z_{t-n}; \gamma) \\ \vdots \\ P_\gamma(z_{T-1}, \dots, z_{T-n}; \gamma) \end{bmatrix} \quad (10)$$

**Loss function.** The first goal of the loss function is to minimize the difference between the predictions $\mathbf{z}$ and $\hat{\mathbf{z}}$ over a batch of size $M$, as $\mathcal{L}_1 = \frac{1}{M}\sum_{i=1}^{M}(z_i - \hat{z}_i)^2$, *i.e.*,

$$\frac{1}{M}\sum_{i=1}^{M}\Big[E_\theta(x_i) - P_\gamma\big(E_\theta(x_{i-1}), \dots, E_\theta(x_{i-n})\big)\Big]^2. \quad (11)$$

One problem with this approach is the convergence to the trivial solution such that $E_\theta(x) = 0\ \forall x$ and $P_\gamma(z) = 0\ \forall z$. To avoid this problem, we induce variance in the encoder's output. Inspired by the VAE [25] we encourage $z_j \in \mathcal{N}(\mu, \sigma^2)$. The values of $\mu$ and $\sigma^2$ effectively define the range of $z$ by renormalization of the metric. As in the VAE [25], we define the second part of the loss function using the Kullback-Leibler divergence (KL-divergence). Thus, $z_j$ is a sample of the random variable $Z \sim \mathcal{N}(\mu_z, \sigma_z^2)$ and we want it to follow a particular prior distribution $Q \sim \mathcal{N}(0, 1)$. Then KL-divergence is given by:

$$\mathcal{L}_2 = \mathrm{KL}(Z||Q) = -\frac{1}{d}\sum_{j=1}^{d}\big(1 + \ln(\sigma_z^2) - \mu_z^2 - \sigma_z^2\big). \quad (12)$$

Here, the KL-divergence is used differently than conventional: VAEs [25] typically assume the encoder finds the correct distribution, and use the sampling trick to obtain the decoder input. In our proposal, we do not sample from the latent distribution. Instead, we constrain the encoder to learn the dynamical state variable. Thus, we calculate the mean $\mu_z$ and variance $\sigma_z^2$ over the batch in the loss.

Finally, combining the two loss functions:

$$\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2 = \frac{1}{M}\sum_{i=1}^{M}(z_i - \hat{z}_i)^2 - \frac{1}{d}\sum_{j=1}^{d}\big(1 + \ln(\sigma_z^2) - \mu_z^2 - \sigma_z^2\big). \quad (13)$$

Our loss function (Eq. 13) serves several purposes: First, $E_\theta(\cdot)$ maps images to random variables $z$ in latent space, and $L_2$ encourages that $z$ is normally distributed. Then, the function $P_\gamma(\cdot)$ preserves the sense of order and relation between latent spaces from different frames. Since we are analyzing a single video, $z$ contains information about the temporal properties of the sequence of frames. Finally, $L_1$ makes the model consistent since the physical predictions made by the physics block $P_\gamma(\cdot)$ are generated by $E_\theta(\cdot)$.

**Training.** For parameter estimation in the physics block $P_\gamma(\cdot)$, we used a learning rate proportional to the initial value $\gamma_i^0$ of the learnable parameter $\gamma_i$, where $\mathrm{lr}_{(\gamma_i)} \sim 10^{[\log_{10}|(\gamma_i^0)|]}$. This approach provides sufficiently large step sizes at the beginning of training to escape local minima. More details about the optimizer and hardware are in the supplementary material.

## 4. Delfys75: a new real-world dataset

Delfys75 contains five experiments: 3 with motion, one with brightness and one with scale dynamics. Motion is overrepresented to enable a deeper comparison with existing methods, which cannot be evaluated on brightness or scale dynamics. Each experiment is recorded in 3 different settings, with five videos per setting, resulting in a dataset of 75 videos. The resolution is $1920 \times 1080$ at 60 frames per second. Sample frames are visualized in Figure 3.

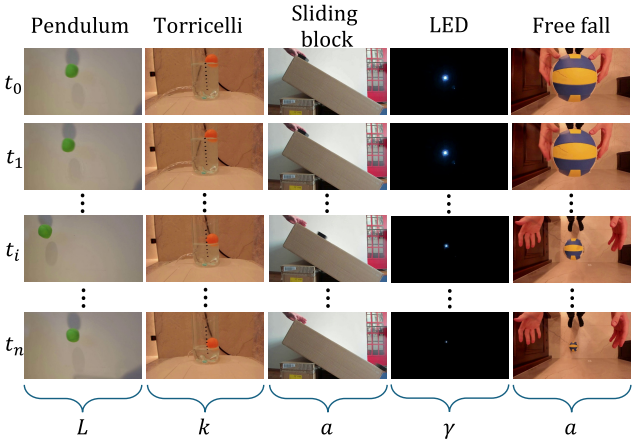



Figure 3. Delfys75 is the first, real-world physical parameter estimation dataset. Top to bottom: first ($t_0$), second ($t_1$), middle ($t_i$), and last frames ($t_n$). Specified at the bottom are the estimated parameters in each scenario. Note the complex shadows, shading, and realistic lighting conditions, in a natural environment.

**Object masks** are binary segmentation masks of the object undergoing the physical transformation. Masks can be learned [20] or otherwise need to be provided [17] with the training data. In our dataset, we provide object masks per frame, annotated using SegmentAnything2 [36], since some baseline methods require it [17]. Our method does not need object masks.

**Parameter ground truths** and estimation errors, are obtained via manual physical parameter estimation. See the supplementary material for details.

### 4.1. Scenarios

The **pendulum** is a classical motion-based system [17, 20, 49] where the position of the pendulum is expressed by the angle $\theta$ and the dynamics are given by:

$$\theta^{(2)} + \zeta\theta^{(1)} + \frac{g}{L}\sin(\theta) = 0. \tag{14}$$

with $\zeta$ being the damping factor, $g$ the gravitational acceleration and $L$ the length of the pendulum. We vary the length of the pendulum to generate the different settings.

The **Torricelli** motion describes the change in water height over time in a container with an orifice through which water is draining. In this setup, a floating ball at the surface serves as an indicator of the water level. Our objective is to determine the constant $k$, related to the water flow rate, from the video as the water level decreases

$$h^{(1)} = k\sqrt{h}. \tag{15}$$

The **sliding block** involves the movement of an object sliding down a ramp [17, 46], with dynamics given by:

$$x^{(2)} = g(\sin(\alpha) - \mu\cos(\alpha)). \tag{16}$$

where $x$ is the position with $\alpha$ the inclination angle of the ramp, $\mu$ the friction coefficient and $g$ is the gravity. Each of our three settings employs a different inclination angle.

In the **LED** experiment, the brightness of an LED lamp is varied. The intensity $I(t)$ of the LED is adjusted following $I(t) = e^{-\gamma t}$ with a controllable decay $\gamma$. We disable camera auto-focus, auto-brightness, and white balancing for this experiment to ensure consistent brightness. $\gamma$ is changed between the three settings.

In the **free fall** scenario, a ball dropped from a height $h_0$ is filmed from above, causing the ball's apparent radius $r(t)$ to decrease as it falls. Using similar triangles, $r(t)$ is described by:

$$r(t) = \frac{r_0 f}{h_0 + \frac{gt^2}{2}}. \tag{17}$$

with $r_0$ the real-life radius of the ball and $f$ the focal length of the camera. We disable camera auto-focus to ensure constant $f$. The radius $r_0$ is modified between settings.

The videos contain hand interactions, which introduce occlusions and variability. The pendulum includes border occlusions and cast shadows. Torricelli has background changes due to water flow, ball rotation, and deformation. The free fall presents perspective distortions. In a a similar way, LED suffers from reflections and non-uniform intensity changes. These factors add complexity and affect the performance of the model.

# 5. Experiments

## 5.1. Synthetic Video Datasets

We start in a fully-controlled setting with three synthetic datasets involving motion, intensity, and scale, visualized in Figure 5. We describe dataset details in the supplementary material. State-of-the-art methods for physical parameter estimation commonly use simulated datasets [10, 20, 41, 43, 49, 50], where objects appear in different colors on a black background. The pendulum system is a typical case of study, but for scale and intensity, there were no baselines available.

**Motion.** We used a pendulum model popular in literature [7, 17, 20]. In this dataset, the state variable is the angle of the pendulum $z_{real} = \theta \in [-180.0°, 180.0°]$.

**Intensity.** We consider time-varying grayscale pixel intensity of an irregular shape. We normalized the intensity dynamics to be in the range $z_{\text{real}} \in [0.2, 1.0]$.

**Scale.** We use a filled circle centered in the middle of the image, where the radius is proportional to the dynamic variable. However, the scaling transformation is not symmetric; while one half of the circle grows, the other half becomes smaller and vice versa. We normalized the range of the radius dynamics between $z_{\text{real}} = r \in [-10, 10]$.

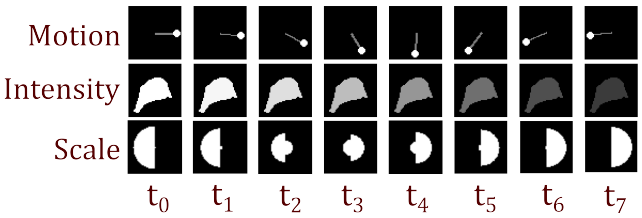


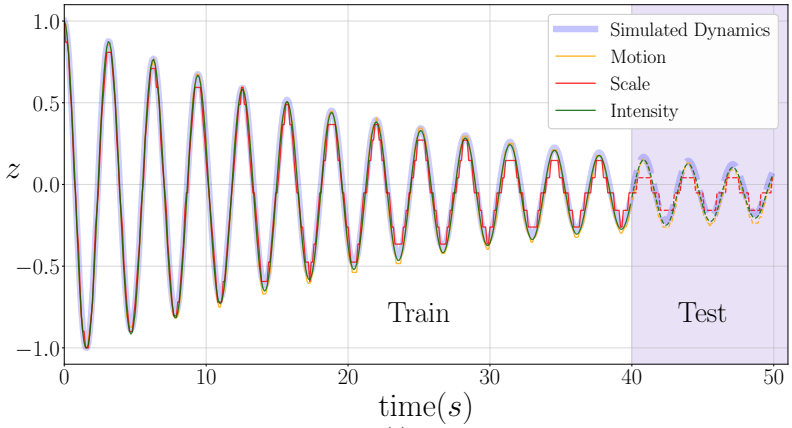

Figure 5. Example frames from the synthetic datasets. Each row shows a different dataset, corresponding to a different continuous dynamical system, and each column a different time sample.

**Dynamics Representation in Latent Space** First, we train our model using the three synthetic datasets and evaluate the dynamics $z$ estimated by the encoder $E_\theta(\cdot)$. We compare $z$ to its ground truth value $z_{real}$, which was used to generate the data. Following Eq. 2, we consider second-order dynamics for all datasets, where the evolution of the state variable $z$ follows a dampened oscillation.

Figure 4 shows our model is capable of estimating the dynamics $z$ for all three datasets. Although understanding oscillations in data can be challenging for neural networks [48], our unsupervised loss fits the dynamical behaviour reasonably, with small deviations from the ground truth.

Importantly, our model has complete physical interpretability since we use the differentiable, second-order ODE (Eq. 2) in the physics block. Due to this physics prior, the model is able to generalize at test time to previously unseen time steps. The network's extrapolation of $z$ to unseen future time steps is shown using dashed lines in Figure 4a.

We observe that the most accurate results are obtained using the 'Intensity' dataset. We attribute this to discretization error: 8-bit inputs of pixel intensity can assume 256 different values. In contrast, with motion and scale, the dynamics are discretized by the pixel locations. In particular, for a $(50 \times 50)$ frame size, the discretization of the dynamic variable is increasingly more impactful as the oscillation amplitude is decreased. This effect is seen clearly in the latent space dynamics of the 'Scale' dataset in Figure 4a (red line), which displays discrete jumps. The discretization error




(a)

| **Equation** | $z^{(2)} + \gamma_1 z^{(1)} + \gamma_0 z = 0$ | |
|---|---|---|
| | $\gamma_0$ | $\gamma_1$ |
| Motion | $3.943 \pm 0.008$ | $0.144 \pm 0.007$ |
| Intensity | $3.887 \pm 0.035$ | $0.089 \pm 0.010$ |
| Scale | $4.055 \pm 0.026$ | $0.910 \pm 0.012$ |
| **GT** | 4.0016 | 0.08 |

(b)

Figure 4. **(a)** Latent space estimation of the dynamic variable $z$ for the three synthetic datasets. The blue line shows the 'ground truth' value $z_{\text{real}}$ of the simulated dynamics. The model was trained with the dynamics of the continuous line while the dashed lines show the ground truth (blue) and predictions (yellow, red, green) on the extrapolated test set. **(b)** Parameter estimation accuracy. Rows 1-3: mean $\pm$ standard deviation of each learnable parameter in the physics block after training, bottom row: ground truth (GT). The values are obtained over 7 different runs with different initializations. We observe good agreement between the predicted and ground truth dynamics.

also disproportionally affects the parameter estimation in the 'Scale' dataset, specifically the prediction of the parameter $\gamma_1$, which is discussed in the next section.

**Parameter Estimation Accuracy** To evaluate our model's accuracy in estimating parameters $\gamma$, we consider second-order equations defined in Eq. 2, where $\gamma_0$ is the frequency and $\gamma_1$ is the damping factor of the oscillations. (See supplement for details.)

Figure 4b shows the $\gamma$ values learned by our model. We find that the model can accurately estimate $\gamma_0$ for all systems. Figure 4a serves as validation that the model indeed fits the correct frequency. The varying accuracy for $\gamma_1$ is attributed to the discretization error discussed in Section 5.1, as the accuracy decrease correlates well with the increased discretization in the motion and scale datasets.

## 5.2. Robustness and Stability

While previous work can reliably generate frame reconstructions using physics blocks in latent space, they often lack an analysis of the parameter estimation. Thus, it is not known if the generated frames correctly use the latent space information in an interpretable manner. In fact, it is known that the baseline models are sensitive to initialization and may fail to converge [17, 33]. In this section, we evaluate the robustness of our model against changes in parameter initializations.

We initialize the learnable parameters $\gamma$ in the interval $[-10.0, 10.0]$ over 7 runs. Ground truth values were $\gamma_0 = 4$ and $\gamma_1 = 0.08$. Figure 6 shows the convergence of the parameter estimation during training with different initializations. While the trajectories may initially diverge, the trained model consistently converges to the ground truth values for each dataset, independent of the initialization. The relatively low standard deviations of the final $\gamma$ estimates in Figure 4b highlights the stability of our model.

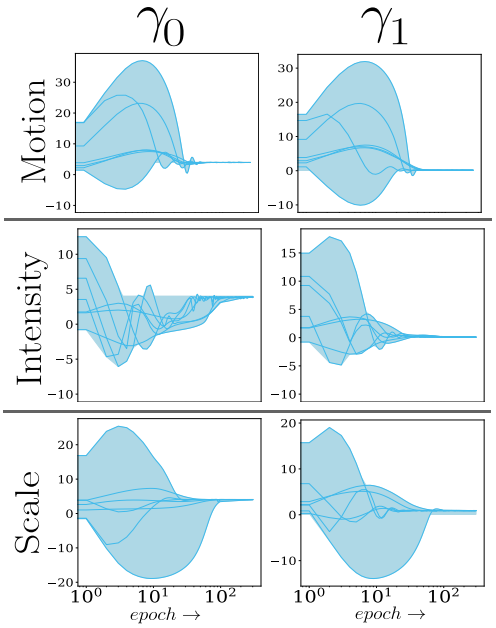



Figure 6. Robustness of the parameter estimation against different initializations. The rows indicate the different dynamical systems, while the columns are the parameters to estimate. The vertical axis corresponds to the parameter value, and the horizontal axis is the epoch. Blue lines show the value of the estimated parameter $\gamma_i$ over training epochs. Since convergence was relatively fast, the horizontal axis is on a logarithmic scale for visibility. The shading highlights the variance of the trajectories before convergence.

## 5.3. Baseline Comparison on Synthetic Data

The baselines PAIG [20] and NIRPI [17] are not designed to handle the intensity and scale settings of our synthetic datasets. Therefore, for a fair comparison, we use the syn-

| | PAIG [20] | PAIG w/o U-Net | NIRPI [17] | Ours |
|---|---|---|---|---|
| Parameters | 5.27M | 4.78M | 75.42K | 4.19M |
| Time per Epoch $[s]$ | 252.72 | 80.56 | 0.11 | 0.95 |
| Decoder | Yes | Yes | Yes | No |
| Mask | No | Yes | Yes | Yes No |

Table 1. Comparison of baseline models evaluated on the public dataset from [20]. Our model only needs a mask if there are multiple objects present in the video.

thetic dataset first proposed in [20] and reused in [17]. It consists of two MNIST digits moving following the spring equation parametrized by the spring constant $k$; the dataset was created using $k = 2$. Details of the governing equation and visualizations are given in the supplement. In this dataset the dynamics is given by the $(x, y)$ position of each digit, therefore the dimensionality of the latent space is $d = 4$.

Table 1 presents a size comparison of the models along with training time per epoch. PAIG [20] employs a U-Net to learn the object masks and does not require them as input. Other baselines, as well as our model for multiple objects, need masked inputs and, therefore, do not employ a segmentation block. We also consider PAIG [20] with masked input and without the U-Net block in our comparisons.

We empirically demonstrate the models' sensitivity to initial conditions, we consider PAIG [20] as proposed by the authors with the U-NET. We train each model twice and initialize the estimated parameter $k$ with values 1.0 and 10.0, respectively. The expected value is $k = 2$ [17, 20]. In Figure 7, we observe that different initializations fail to converge to the correct value of $k$ for the baselines, while our model is consistent and converges to the desired value accurately. This experiment demonstrates that our model can also successfully tackle problems with multiple objects.

### 5.4. Real-world video experiments on Delfys75

We evaluate our model and the baselines on Delfys75 dataset and dynamics described in Section 4. Our model was trained without masks, while PAIG [20] needs to learn them, and NIRPI [17] requires them as input. For both baselines, we used the hyperparameters given in their respective papers. We trained our model for 500 epochs to ensure parameters converged for all experiments; the learning rate was chosen as described in Section 3. Further training details and an analysis of the latent space of our model are given in the supplement. Our model uses a prior $Q \sim \mathcal{N}(0, 1)$ for $\mathcal{L}_2$ (Eq. 12), except for the Torricelli experiment, where the governing equation (Eq. 15) includes a square root that introduces conflict if $z < 0$. To avoid this, we change the

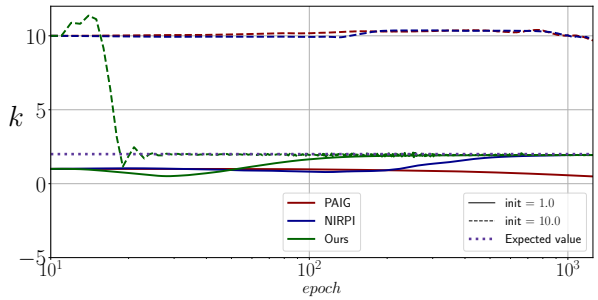



Figure 7. Robustness of the parameter estimation compared to baselines. For a fair comparison, we use the synthetic dataset created originally by the authors of the baseline papers to evaluate their models [20]. We plot the trajectories of the estimated parameter $k$ during training with different initializations for our model (green) and for the two baseline models (red, blue). Dotted lines correspond to an initial value of $k = 10.0$, and solid lines to $k = 1.0$. Our model converges robustly to the ground truth value of $k = 2.0$.

expected prior to $Q \sim \mathcal{N}(1, 0.2)$. As explained in Section 3, this change leads to a different renormalization factor.

Each model was tested on five videos to compute the mean and standard deviation for each setting. For the pendulum and the LED, the parameter of each setting is estimated. For the sliding block, multiple values of the parameters result in the same dynamics; hence only the total acceleration $a = g(\sin(\alpha) - \mu\cos(\alpha))$ is estimated. The gravitational constant is also predicted in the dropped ball experiment. Baselines are excluded from intensity and scale experiments as they only support motion dynamics. Table 2 shows the ground truth (GT) and the learned value of each parameter.

On all settings, our proposed model performs parameter estimation relatively accurately compared to baselines. In particular, we observe that PAIG [20] estimates parameters only close to the initial value (1.0). We train all models on a single video at a time on each setting. This leads to a small training set compared to the 10,000 videos used in [20], illustrating the data efficiency of our model compared to PAIG [20]. On the other hand, parameter estimates for NIRPI [17] converge to varying values, but are on average lower accuracy than those fitted by the proposed model.

## 6. Discussion and Limitations

We present a novel model and dataset for unsupervised physical parameter estimation from video. While previous methods do not study phenomena other than motion, we go beyond motion and include a variety of dynamical systems.

The proposed model avoids frame prediction, which is challenging, but useful for visualization. However, decoder-based methods need to be designed for the particular dynamical system, as a transformation-agnostic decoder may struggle with high-quality reconstruction. Without decoders, our approach works across various dynamical systems.

In addition, we directly examine latent space predictions, unlike prior models [17, 20, 23], which focus on frame re-

| | Pendulum $L\,[m]$ | | | Torricelli $k\left[\frac{\sqrt{m}}{s^2}\right]$ | | | Sliding block $a\left[\frac{m}{s^2}\right]$ | | |
|---|---|---|---|---|---|---|---|---|---|
| PAIG | $1.01 \pm 0.03$ | $1.01 \pm 0.04$ | $1.01 \pm 0.04$ | $0.99 \pm 0.01$ | $0.99 \pm 0.01$ | $0.97 \pm 0.02$ | $0.35 \pm 0.03$ | $0.38 \pm 0.02$ | $0.37 \pm 0.04$ |
| NIRPI | $0.77 \pm 0.33$ | $0.84 \pm 0.53$ | $0.63 \pm 0.38$ | $0.21 \pm 0.03$ | $0.14 \pm 0.04$ | $0.16 \pm 0.01$ | $-0.09 \pm 0.88$ | $-0.01 \pm 0.5$ | $-0.06 \pm 0.01$ |
| Ours | $0.51 \pm 0.01$ | $1.07 \pm 0.2$ | $1.30 \pm 0.02$ | $0.0094 \pm 4e^{-4}$ | $0.0132 \pm 5e^{-4}$ | $0.0167 \pm 4e^{-4}$ | $1.29 \pm 0.1$ | $2.70 \pm 0.09$ | $3.44 \pm 0.19$ |
| **GT** | 0.45 | 0.90 | 1.50 | 0.0095 | 0.0128 | 0.0162 | 1.441 | 2.300 | 3.141 |

(a)

| | LED $\gamma$ | | | Free fall scale $a\left[\frac{m}{s^2}\right]$ | | |
|---|---|---|---|---|---|---|
| | | | | Small ball | Medium ball | Large ball |
| Ours | $2.24 \pm 0.36$ | $0.97 \pm 0.04$ | $0.41 \pm 0.04$ | $15.0 \pm 2.1$ | $9.51 \pm 1.27$ | $10.22 \pm 1.21$ |
| **GT** | 2.3 | 0.92 | 0.46 | 9.8 | 9.8 | 9.8 |

(b)

Table 2. Parameter estimation accuracy on real-world videos. The table shows how closely each model estimates the ground truth (GT) parameter values, with mean and standard deviation calculated over five videos per setup. On **(a)** we show the so-called motion problems and compare against the baselines. On **(b)** since there are no baseline methods for non-motion scenarios, and thus, we could not evaluate baselines for the LED and free-fall scale videos. Highlighted values indicate the estimates that are closest to the expected values.

construction and do not discuss the accuracy of predicted dynamics. Since parameter estimation is the main goal, reconstruction is simply a tool to define the unsupervised loss, and therefore, the latent space should be analysed closely.

Our model does not resolve the absolute scale of the state variable, for example, that the pendulum goes from $[-180.0°, 180.0°]$. Yet, thanks to the loss function, the model solves its own metric, ensuring assumptions made in Section 3. The choice of the prior target distribution simply normalizes the dynamics and should not affect performance, as seen in Section 5.4 with the Torricelli experiment. Baselines implicitly or explicitly do this normalization using the spatial transformer, forcing the prediction to be in the pixel metric. In contrast, all our dynamics depend on the prior distribution assumed by the KL-divergence.

Baseline methods are sensitive to $\gamma$ initialization. Our method may also converge to local minima with small learning rates and high initial parameters. Therefore, we enforce variance in the learning rate depending on the initial values of the parameters, which allow for wider search spaces.

Our model demonstrates good performance on real-world videos, successfully handling challenges such as shadows and perspective distortions, without the need for object masks. Additionally, our approach outperforms baseline models–which prioritize reconstruction tasks–on parameter learning. Using a spatial transformer, baselines rely on the positions and velocities predicted by the encoder, overlooking minor variations introduced by the governing equation.

Our new real-world dataset is versatile: While baseline models do not support non-motion dynamics and require object masks, they could still be trained on the motion-based subset of our proposed dataset with the accompanying masks. Thus, we believe our dataset will be helpful for evaluating future models on a common dataset, enabling more comprehensive comparisons in complex real-world settings.

**Limitations.** Our model is suited for continuous, autonomous differential equations. However, some systems, such as fluids, are described with more complex differential equations. In addition, we need to guarantee that Eq. 7 is differentiable. Taken together, our model needs to be extended to tackle more complex use cases, including multiple objects in a scene with independent dynamics.

Due to discretization, performance may vary with resolution. For example, small changes of the dynamic variable in the scale dataset were challenging to detect as the model cannot achieve sub-pixel accuracy.

We avoid predicting object masks in videos. Real-world experiments in Section 5.4 show that our model does not need masks when trained with complex backgrounds, whereas baseline methods still require them [17, 20]. However, when multiple objects interact, masks become necessary, as demonstrated in experiment 5.3. While learning masks is an effective solution [18, 20], it relies heavily on the spatial transformer limited to roto-translation problems.

Nevertheless, limitations are mostly related to the early stages of the problem, opening multiple opportunities for future works; with our work, we provide a base which can be extended to more complex problems in different scientific domains and initiate data sharing to evaluate future models.

## References


[1] Claire Adam-Bourdarios, Glen Cowan, Cecile Germain-Renaud, Isabelle Guyon, Balázs Kégl, and David Rousseau. The higgs machine learning challenge. In *Journal of Physics: Conference Series*, page 072015. IOP Publishing, 2015. 1, 2

[2] Mary P Anderson, W Woessner, and Randall J Hunt. Chapter 9-model calibration: assessing performance. *Applied groundwater modeling*, pages 375–441, 2015. 1

[3] Martin Asenov, Michael Burke, Daniel Angelov, Todor Davchev, Kartic Subr, and Subramanian Ramamoorthy. Vid2param: Modeling of dynamics parameters from video. *IEEE Robotics and Automation Letters*, 5(2):414–421, 2019. 1

[4] Nicholas M Ball and Robert J Brunner. Data mining and machine learning in astronomy. *International Journal of Modern Physics D*, 19(07):1049–1106, 2010. 1, 2

[5] Ershad Banijamali, Rui Shu, Hung Bui, Ali Ghodsi, et al. Robust locally-linear controllable embedding. In *International Conference on Artificial Intelligence and Statistics*, pages 1751–1759. PMLR, 2018. 2

[6] Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, et al. Interaction networks for learning about objects, relations and physics. *Advances in neural information processing systems*, 29, 2016. 2

[7] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016. 1, 2, 5

[8] Kathleen Champion, Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451, 2019. 1, 2

[9] Miles Cranmer, Sam Greydanus, Stephan Hoyer, Peter Battaglia, David Spergel, and Shirley Ho. Lagrangian neural networks. *arXiv preprint arXiv:2003.04630*, 2020. 2

[10] Filipe de Avila Belbute-Peres, Kevin Smith, Kelsey Allen, Josh Tenenbaum, and J Zico Kolter. End-to-end differentiable physics for learning and control. *Advances in neural information processing systems*, 31, 2018. 2, 5

[11] Katerina Fragkiadaki, Pulkit Agrawal, Sergey Levine, and Jitendra Malik. Learning visual predictive models of physics for playing billiards. *arXiv preprint arXiv:1511.07404*, 2015. 2

[12] William D Fries, Xiaolong He, and Youngsoo Choi. Lasdi: Parametric latent space dynamics identification. *Computer Methods in Applied Mechanics and Engineering*, 399:115436, 2022. 2

[13] Han Gao, Matthew J Zahr, and Jian-Xun Wang. Physics-informed graph neural galerkin networks: A unified framework for solving pde-governed forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering*, 390:114502, 2022. 2

[14] Craig Gin, Bethany Lusch, Steven L Brunton, and J Nathan Kutz. Deep learning models for global coordinate transformations that linearise pdes. *European Journal of Applied Mathematics*, 32(3):515–539, 2021. 2

[15] Vincent Le Guen and Nicolas Thome. Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11474–11484, 2020. 2

[16] Ryan N Gutenkunst, Joshua J Waterfall, Fergal P Casey, Kevin S Brown, Christopher R Myers, and James P Sethna. Universally sloppy parameter sensitivities in systems biology models. *PLoS computational biology*, 3(10):e189, 2007. 1

[17] Florian Hofherr, Lukas Koestler, Florian Bernard, and Daniel Cremers. Neural implicit representations for physical parameter inference from a single video. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2093–2103, 2023. 1, 2, 3, 4, 5, 6, 7, 8

[18] Jun-Ting Hsieh, Bingbin Liu, De-An Huang, Li F Fei-Fei, and Juan Carlos Niebles. Learning to decompose and disentangle representations for video prediction. *Advances in neural information processing systems*, 31, 2018. 2, 3, 8

[19] Raban Iten, Tony Metger, Henrik Wilming, Lídia Del Rio, and Renato Renner. Discovering physical concepts with neural networks. *Physical review letters*, 124(1):010508, 2020. 1, 2

[20] Miguel Jaques, Michael Burke, and Timothy Hospedales. Physics-as-inverse-graphics: Unsupervised physical parameter estimation from video. *In International Conference on Learning Representations*, 2020. 1, 2, 3, 4, 5, 6, 7, 8

[21] Miguel Jaques, Michael Burke, and Timothy M Hospedales. Newtonianvae: Proportional control and goal identification from pixels via physical latent spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4454–4463, 2021. 2

[22] Navami Kairanda, Edith Tretschk, Mohamed Elgharib, Christian Theobalt, and Vladislav Golyanik. f-sft: Shape-from-template with a physics-based deformation model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3948–3958, 2022. 2

[23] Rama Kandukuri, Jan Achterhold, Michael Moeller, and Joerg Stueckler. Learning to identify physical parameters from video using differentiable physics. In *DAGM German conference on pattern recognition*, pages 44–57. Springer, 2020. 2, 7

[24] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021. 2

[25] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *ICLR*, 2014. 2, 4

[26] Jannik Kossen, Karl Stelzner, Marcel Hüßing, Claas Voelcker, and Kristian Kersting. Structured object-aware physics prediction for video modeling and planning. In *International Conference on Learning Representations (ICLR)*, 2020. 2

[27] Kookjin Lee, Nathaniel Trask, and Panos Stinis. Structure-preserving sparse identification of nonlinear dynamics for data-driven modeling. In *Mathematical and Scientific Machine Learning*, pages 65–80. PMLR, 2022. 2

[28] Attila Lengyel, Sourav Garg, Michael Milford, and Jan C van Gemert. Zero-shot day-night domain adaptation with a physics prior. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4399–4409, 2021. 2

[29] Maxwell W Libbrecht and William Stafford Noble. Machine learning applications in genetics and genomics. *Nature Reviews Genetics*, 16(6):321–332, 2015. 1, 2

[30] Michael Lutter, Christian Ritter, and Jan Peters. Deep lagrangian networks: Using physics as model prior for deep learning. 2019. 2

[31] Erik Meijering, Anne E Carpenter, Hanchuan Peng, Fred A Hamprecht, and Jean-Christophe Olivo-Marin. Imagining the future of bioimage analysis. *Nature biotechnology*, 34(12): 1250–1255, 2016. 1, 2

[32] Zeng Meng, Qiaochu Qian, Mengqiang Xu, Bo Yu, Ali Rıza Yıldız, and Seyedali Mirjalili. Pinn-form: A new physics-informed neural network for reliability analysis with partial differential equation. *Computer Methods in Applied Mechanics and Engineering*, 414:116172, 2023. 2

[33] Jaques Miguel. Physics-as-inverse-graphics. https://github.com/seuqaj114/paig, 2019. 6

[34] Christopher Rackauckas, Yingbo Ma, Julius Martensen, Collin Warner, Kirill Zubov, Rohit Supekar, Dominic Skinner, Ali Ramadhan, and Alan Edelman. Universal differential equations for scientific machine learning. *arXiv preprint arXiv:2001.04385*, 2020. 2

[35] M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. 2

[36] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025. 4

[37] Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *science*, 324(5923):81–85, 2009. 1

[38] Naoya Takeishi and Alexandros Kalousis. Physics-integrated variational autoencoders for robust and interpretable generative modeling. *Advances in Neural Information Processing Systems*, 34:14809–14821, 2021. 2

[39] Peter Toth, Danilo J. Rezende, Andrew Jaegle, Sébastien Racanière, Aleksandar Botev, and Irina Higgins. Hamiltonian generative networks. In *International Conference on Learning Representations (ICLR)*, 2020. 2

[40] Gaël Varoquaux and Veronika Cheplygina. Machine learning for medical imaging: methodological failures and recommendations for the future. *NPJ digital medicine*, 5(1):48, 2022. 1, 2

[41] Petar Veličković, Matko Bošnjak, Thomas Kipf, Alexander Lerchner, Raia Hadsell, Razvan Pascanu, and Charles Blundell. Reasoning-modulated representations. In *Learning on Graphs Conference*, pages 50–1. PMLR, 2022. 2, 5

[42] Manuel Watter, Jost Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. *Advances in neural information processing systems*, 28, 2015. 2

[43] Nicholas Watters, Daniel Zoran, Theophane Weber, Peter Battaglia, Razvan Pascanu, and Andrea Tacchetti. Visual interaction networks: Learning a physics simulator from video. *Advances in neural information processing systems*, 30, 2017. 1, 2, 5

[44] Sebastian Weiss, Robert Maier, Daniel Cremers, Rudiger Westermann, and Nils Thuerey. Correspondence-free material reconstruction using sparse surface constraints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4686–4695, 2020. 2, 3

[45] Jiajun Wu, Ilker Yildirim, Joseph J Lim, Bill Freeman, and Josh Tenenbaum. Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. *Advances in neural information processing systems*, 28, 2015. 2

[46] Jiajun Wu, Joseph J Lim, Hongyi Zhang, Joshua B Tenenbaum, and William T Freeman. Physics 101: Learning physical object properties from unlabeled videos. In *British Machine Vision Conference*, 2016. 2, 5

[47] Jiajun Wu, Erika Lu, Pushmeet Kohli, Bill Freeman, and Josh Tenenbaum. Learning to see physics via visual de-animation. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. 2

[48] Zhi-Qin John Xu, Yaoyu Zhang, Tao Luo, Yanyang Xiao, and Zheng Ma. Frequency principle: Fourier analysis sheds light on deep neural networks. *Communications in Computational Physics*, 28(5):1746–1767, 2020. 5

[49] Tsung-Yen Yang, Justinian Rosca, Karthik Narasimhan, and Peter J Ramadge. Learning physics constrained dynamics using autoencoders. *Advances in Neural Information Processing Systems*, 35:17157–17172, 2022. 2, 4, 5

[50] David Zheng, Vinson Luo, Jiajun Wu, and Joshua B. Tenenbaum. Unsupervised learning of latent physical properties using perception-prediction networks. In *Proceedings of the 34th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 1–10. AUAI Press, 2018. 2, 5

[51] Yaofeng Desmond Zhong and Naomi Leonard. Unsupervised learning of lagrangian dynamics from images for prediction and control. In *Advances in Neural Information Processing Systems*, pages 10741–10752. Curran Associates, Inc., 2020. 2