Neural Implicit Representations for Physical Parameter Inference from a Single Video

Florian Hofherr[1] Lukas Koestler[1] Florian Bernard[2] Daniel Cremers[1]
[1]Technical University of Munich [2]University of Bonn

Abstract

Neural networks have recently been used to analyze diverse physical systems and to identify the underlying dynamics. While existing methods achieve impressive results, they are limited by their strong demand for training data and their weak generalization abilities to out-of-distribution data. To overcome these limitations, we propose to combine neural implicit representations for appearance modeling with neural ordinary differential equations (ODEs) for modelling planar physical phenomena to obtain a dynamic scene representation that can be identified directly from visual observations. Our proposed model combines several unique advantages: (i) Contrary to existing approaches that require large training datasets, we are able to identify physical parameters from only a single video. (ii) The use of neural implicit representations enables the processing of high-resolution videos and the synthesis of photo-realistic images. (iii) The embedded neural ODE has a known parametric form that allows for the identification of interpretable physical parameters, and (iv) long-term prediction in state space. (v) Furthermore, the photo-realistic rendering of novel scenes with modified physical parameters becomes possible.

1. Introduction

For many physical phenomena, humans are able to infer (a rough estimation of) physical quantities from observing a scene, and are even capable to predict what is going to happen in the (near) future. In contrast, physical understanding from videos is an open problem in machine learning. The physics of many real-world phenomena can be described concisely and accurately using differential equations. However, such equations are usually formulated in terms of abstracted quantities that are typically not directly observable using commodity sensors, such as cameras. For example, the dynamics of a pendulum are described by the deflection angle and the angular velocity as the time-varying state and the damping coefficient, and the pendulum's length as parameters. Automatically extracting those physical parameters directly from video data is challenging. Due to the difficulties in direct observation of those quantities in images, and their complex relationship with the physical process, measuring such quantities in experiments often necessitates a trained expert operating customised equipment.

Recently, the combination of deep learning and physics has become popular, particularly in the context of video prediction. While earlier works [30, 16, 42, 11, 58, 10, 20, 43] require coordinate data, i.e. abstracted physical quantities that are not directly accessible from the video, more recent works directly use image data [49, 12, 22, 24, 52, 28, 59, 26, 50]. A major downside of all these approaches is, that they rely on massive amounts of training data, and exhibit poor generalization abilities if the observation deviates from the training distribution, as we experimentally confirm. In contrast, our proposed combination of a parametric dynamics model with a neural scene representation allows for identification of the dynamics from only a single high resolution video. Also, due to our per-scene approach, our method does not suffer from generalization issues either.

Several of the previously mentioned works model physical systems using Lagrangian or Hamiltonian energy formulations [30, 16, 11, 10, 52, 58, 28, 59], or other general physics models [26]. While those are a elegant approaches that allow the model to adapt to different physical systems, they have two drawbacks. First, the general models are part of the reason why large amounts of data are required. Second, once the system dynamics have been identified, they are not easily interpretable. Questions like "*How would the scene look like if we double the damping*" cannot be answered. In contrast, we estimate physically meaningful parameters of the underlying dynamics like the length of a pendulum or the friction coefficient of a sliding block. We find experimentally that using an ODE-based dynamics model gives more accurate long-term predictions. Moreover, due to the combination with the photo-realistic rendering capacities of our neural appearance representation, we are able to re-render the scene with adapted parameters.

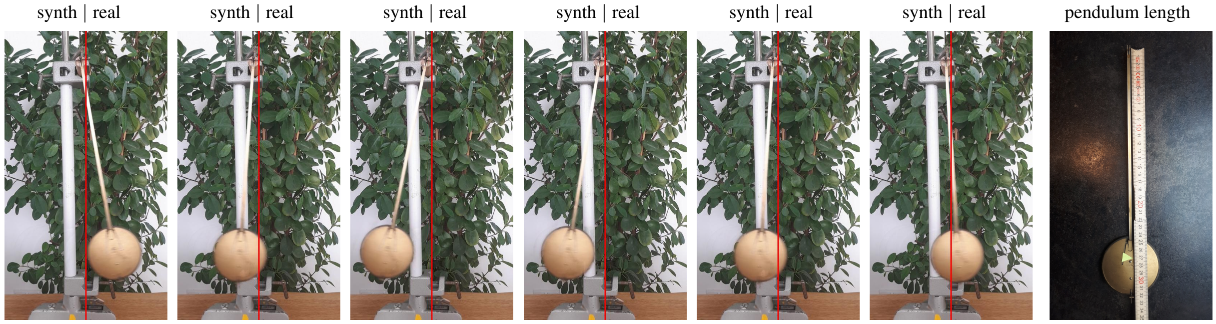

Figure 1: Our method infers physical parameters directly from real-world videos, like the shown pendulum motion. Separated by the red line, the right half of each image shows the input frame, and the left half shows our reconstruction based on physical parameters that we estimate from the input. We show 6 out of 10 frames that were used for training. The proposed model can precisely recover the metric length of the pendulum from the monocular video (relative error to true length is less than 4.1%). Best viewed on screen with magnification. Please also consider the supplementary video.

We summarize our main contributions as follows:

1. We present the first method that uses neural implicit representations to identify physical parameters for planar dynamics from a single video.
2. Our approach infers parameters of an underlying ODE-based physical model that directly allows for interpretability and long-term predictions.
3. The unique combination of powerful neural implicit representations with rich physical models allows to synthesize high-resolution and photo-realistic imagery. Moreover, it enables physical editing by rendering novel scenes with modified physical parameters.
4. Contrary to existing learning-based approaches that require large corpora of training data, we propose a *per-scene* model, so that only a single short video clip that depicts the physical phenomenon is necessary.

See **https://florianhofherr.github.io/phys-param-inference** and the supplement for architecture & training details and the supplementary video. This work is of fundamental character and thus has less immediate potential for negative societal impact. We discuss this in detail in the supplement.

2. Related Work

The combination of machine learning and physics has been addressed across a broad range of topics. Machine learning was used to aid physics research [4, 27], and physics was used within machine learning models, e.g. for automatic question answering from videos [8, 3]. A great overview over physics-informed machine learning can be found in [25]. In this work we focus specifically on extracting physical models from videos, so that we discuss related works that we consider most relevant in this context.

Physical dynamics in the context of learning. While neural networks have led to remarkable results across diverse domains, the inference and representation of physical principles like energy conservation, is still a challenge in the context of learning and requires additional constraints. Generalized energy functions are one way to endow models with physics-based priors. For example, [16, 10] and [52] use a neural network to parameterize the Hamiltonian of a system, which relates the total energy to the change of the state. This approach allows to infer the dynamics of systems with conserved energy, like an undamped pendulum. [47] augment the Hamiltonian by a learned Rayleigh dissipation function to model systems that do not conserve energy, which are more common in the real world [15].

One disadvantage of the Hamiltionian is that *canonical coordinates* need to be used. To eliminate this constraint, other works use the Lagrangian to model the energy of the system. Since this formalism is more complex, [30] and [59] restrict the Lagrangian to the case of rigid-body dynamics to model systems with multiple degrees of freedom, such as a pole on a cart, or a robotic arm. [11] use a neural network to parameterize a general Lagrangian to infer the dynamics of a relativistic particle in a uniform potential.

Another problem of many previous approaches is that they do not allow for interpretation of individual learned system parameters. For example, [18] learns dynamics in the form of a general PDE in a latent space, which, like the aforementioned works based on energy functions, prohibits interpretation of the learned physical model (e.g in the form of interpretable parameters). In contrast, there are also approaches that explicitly incorporate the underlying dynamics into learning frameworks. [22] unroll the Euler integration of the ordinary differential equation of bouncing balls, as well as balls connected by a spring, to identify the physical parameters like the spring constant. [24] and [12] propose to use a linear complementarity problem to differentiably simulate rigid multi-body dynamics that can also handle object interaction and friction. [41] and [40] add uncertainty propagation by combining numeric stepping schemes with Gaussian processes. For our method, we also rely on the advantages of modelling the underlying physics explicitly to obtain interpretable parameter estimates.

Inferring physical properties from video. While many approaches work with trajectories in state space, there are also several works that operate directly on videos. In this case, the information about physical quantities is substantially more abstract, so that uncovering dynamics from video data is a significantly more difficult problem. In their seminal work [54] consider objects sliding down a plane. By tracking the objects, they estimate velocity vectors that are used to supervise a rigid body simulation of the respective object. Similarly, [21] track the trajectories of keypoints for more complex rigid body motions like a bouncing ball, and estimate the physical parameters and the most likely model from a family of possible models by comparing the tracked trajectory to the projection of a simulated 3D trajectory. Both methods rely on correctly identifying the object tracks and do not use the rich information contained in the image directly. Also, video extrapolation is not easily possible. [53] and [23] consider deformable objects and solve a partial differential equation to simulate the deformations. Wile the first method uses depth values as supervision, the second one employs a photometric loss. [14] extract vibration modes from a video and identify the material parameters by comparing to the eigenmodes of the object mesh. While those methods show impressive results, all three require a 3D template mesh as additional information, which may limit their practical applicability.

More recently, several end-to-end learning approaches have been proposed. [26] combine the state prediction of an LSTM from an image with the prediction of a graph neural network from the previous state to propagate the state in time. Using the Sum-Product Attend-Infer-Repeat (SuPAIR) model ([48]) they render images from the state predictions and use the input image sequence as supervision. [12, 22] and [24] use an encoder to extract the initial state of several objects from the combination of images and object masks. After propagating the physical state over time, they use carefully crafted decoders to transform the state back into images to allow for end-to-end training. [59] and [52] use a variational autoencoder (VAE) to predict posterior information about the initial state and combine this with an energy based representation of the dynamics and a final decoding stage. [50] improve the VAE based approach by using known explicit physical models as prior knowledge. [6] combine Mask R-CNN [19] with a VAE to predict symbolic equations. All of these approaches require large amounts of data to train the complex encoder and decoder modules. In contrast, our approach does not rely on trainable encoder or decoder structures. Instead it combines neural implicit representations to model the scene appearance with the estimation of the parameters of a known, parameteric ODE, and is able to infer physical models from a single video.

Implicit representations. Recently, neural implicit representations have gained popularity due to their theoretical elegance and performance in novel view synthesis. The idea is to use a neural network to parametrize a function that maps a spatial location to a spatial feature. For example occupancy values [31, 9, 38], or signed distance functions [36, 17, 1] can be used to represent geometric shapes. In the area of multiview 3D surface reconstruction as well as novel view synthesis, a representation for density or signed distance, is combined with neural color fields to represent shape and appearance [45, 32, 56, 34, 2]. To model dynamic scenes, there have been several approaches that parametrize a displacement field and model the scene in a reference configuration [33, 37, 39]. On the other hand, several approaches [55, 29, 13] include the time as an input to the neural representation and regularize the network using constraints based on appearance, geometry, and pretrained depth or flow networks – however, none of these methods uses physics-based constraints, e.g. by enforcing Newtonian motion. An exeption is the work by Song et al. that use the solution of an ODE as regularization of a motion network to crate dynamic NeRFs [46]. In contrast to our work, this approach does not enforce the physics to be exact. While the majority of works on implicit representations focuses on shape, [44] show the generality of implicit representations by representing images and audio. We combine such representations with explicit physical models.

3. Estimating Physical Models with Neural Implicit Representations

Our main goal is the estimation of physical parameters from a single video. We focus on the setting of a static camera, a static background, and rigid objects that are moving according to some physical phenomenon and exhibit a motion that can be restricted to a plane. We model the dynamics of the objects using an ordinary differential equation (ODE) and use implicit neural representations to model the appearance, where the static background and the planar dynamics allow us to model the appearance in 2D. Our objective is to estimate the unknown physical parameters, and the initial conditions, of the ODE, as well as the parameters of the appearance representations. For estimating these quantities directly from an input video, we utilise a photometric loss that imposes similarity between the generated frames and the input. Due to the parametric dynamics model and the photorealistic appearance representation, we can use the result also as a generative model to render videos with varying physical parameters. We would like to note that neural radiance fields have shown convincing performance in 3D and hence the proposed method is a promising step towards physical parameter estimation in three dimensions.

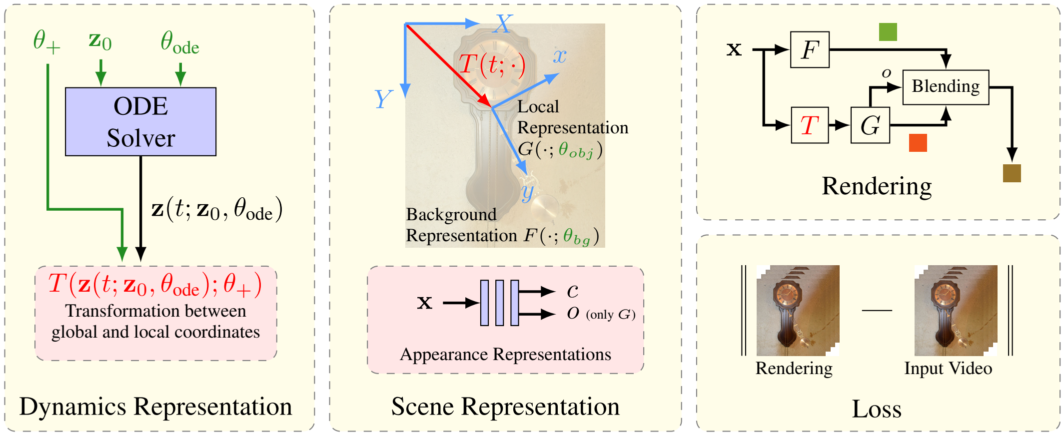

Figure 2: Overview of our approach. **Dynamics Representation:** The dynamics in the video are modelled by an ordinary differential equation (ODE), which is solved depending on unknown initial conditions $\mathbf{z}_0$ and unknown physical parameters θ_{ode}. The solution curve $\mathbf{z}\left(t;\mathbf{z}_0,\theta_{\text{ode}}\right)$ is used to parametrize a time-dependent transformation $T(\mathbf{z}\left(t;\mathbf{z}_0,\theta_{\text{ode}}\right),\theta_+)$ from the global coordinates XY of the background to the local coordinates xy of the moving object. The (unknown) parameters θ_+ encode additional degrees of freedom of the transformation, for example the pivot point of a pendulum. **Scene Representation:** The functions $F(\cdot;\theta_{\text{bg}})$ and $G(\cdot;\theta_{\text{obj}})$ are neural networks that model the appearance of the background and of the object, using color c and opacity o (only for the foreground objects). **Rendering:** Rendering is done by blending the foreground and the background color based on the opacity of the foreground objects. **Loss:** We can estimate the unknown physical parameters for a given video based on a rendering loss which penalizes the discrepancy between the input video frames and the rendered video. All estimated parameters and network weights are shown in green text in the figure.

3.1. Modeling the Dynamics

For most of the dynamics that can be observed in nature, the temporal evolution of the state can be described by an ODE. For example, for a pendulum the state variables are the deflection angle and the angular velocity, and a two dimensional ODE can be used to describe the dynamics.

In general, we write $\dot{\mathbf{z}} = f\left(\mathbf{z},t;\theta_{ode}\right)$ to describe the ODE[1], where $\mathbf{z}\in\mathbb{R}^n$ denotes the state variable, $t\in\mathbb{R}$ denotes time and $\theta_{ode}\in\mathbb{R}^m$ are the unknown physical parameters. Using the initial conditions $\mathbf{z}_0\in\mathbb{R}^n$ at the initial time t_0, we can write the solution of the ODE as

$$\mathbf{z}\left(t;\mathbf{z}_0,\theta_{ode}\right) = \mathbf{z}_0 + \int_{t_0}^{t} f\left(\mathbf{z}(\tau),\tau;\theta_{ode}\right)\mathrm{d}\tau. \quad (1)$$

Note that the solution curve $\mathbf{z}\left(t;\mathbf{z}_0,\theta_{ode}\right)\subset\mathbb{R}^n$ depends both on the unknown initial conditions $\mathbf{z}_0$, as well as on the unknown physical parameters θ_{ode}.

In practice, the solution to Equation (1) is typically approximated by numeric integration. In our context of physical parameter estimation from videos, we build upon [7], who proposed an approach to compute gradients of the solution curve of an ODE with respect to its parameters. With that, it becomes possible to differentiate through the solution in Equation (1) and therefore we can use gradient-based methods to estimate $\mathbf{z}_0$ and θ_{ode}.

[1] W.l.o.g. we consider first-order ODEs here, since it is possible to reduce the order to one by introducing additional state variables.

3.2. Differentiable Rendering of the Video Frames

To render the video frames, we draw inspiration from the recent advances in neural implicit representations. To this end, we combine one representation for the static background with a representation for appearance and shape of dynamic foreground objects. By composing the learned background with the dynamic foreground objects, whose poses are determined by the solution of the ODE encoding the physical phenomenon, we obtain a dynamic representation of the overall scene. Doing so allows us to query the color values on a pixel grid, so that we are able to render video frames in a differentiable manner (cf. Figure 2).

Representation of the background. The appearance of the static background is modeled by a function $F(\cdot;\theta_{\text{bg}})$ that maps a 2D location $\mathbf{x}$ to a color value $\mathbf{c}\in\mathbb{R}^3$. We use a neural network with learnable parameters θ_{bg} to represent $F(\cdot;\theta_{\text{bg}})$. To improve the ability of the neural network to learn high frequency variations in appearance, we use Fourier features [51] that map the input $\mathbf{x}\in\mathbb{R}^2$ to a vector $\gamma\left(\mathbf{x}\right)\in\mathbb{R}^{N_{\text{Fourier}}}$, where N_{Fourier} is the numbers Fourier features used. The full representation of the background then reads $c_{\text{bg}}\left(\mathbf{x}\right) = F(\gamma\left(\mathbf{x}\right);\theta_{\text{bg}})$. For a more detailed discussion of the architecture, we refer to the supplement.

Representation of dynamic objects. To compose the static background and the dynamically moving objects into the full scene, we draw inspiration from both [35] and [57], who use implicit representations to decompose a dynamic 3D scene into a background representation and dynamically moving local representations. A drawback of their works is that they do not use physical dynamics models to constrain the dynamics, and therefore require strong supervisory signals like the trajectories and the dimensions of the bounding boxes in the first case or data from a multi-camera rig in the second case. In contrast, we use the transformation $T_t = T\left(\mathbf{z}\left(t; \mathbf{z}_0, \theta_{\text{ode}}\right), \theta_+\right)$ that is parametrized by the simulation of a physical phenomenon to position the dynamically moving local representation in the overall scene. Besides the unknown initial condition $\mathbf{z}_0$ and the physical parameters θ_{ode} of the ODE, we can use additional parameters θ_+ for the parametrization. In case of the pendulum $\mathbf{z}_0$ are initial angle and anglular velocity, θ_{ode} contains the length and the damping and θ_+ is the pivot point of a pendulum. See the supplement for more details. T_t is a time dependent, affine 2D transformation that maps from global to local coordinates and therefore can model a movements of the object in a plane that is parallel to the (static) camera.

Similarly to the background, the appearance of each individual dynamic object is modeled in terms of an implicit neural representation (in the local coordinate system). In contrast to the background, we augment the color output $c \in \mathbb{R}^C$ of the dynamic object representation with an additional opacity value $o \in [0, 1]$, which allows us to model objects with arbitrary shape. We write the representation of a dynamic object in the global coordinate system as $\left(c_{\text{obj}}\left(\mathbf{x}\right), o\left(\boldsymbol{x}\right)\right) = G(\gamma\left(\boldsymbol{x}'\right); \theta_{\text{obj}})$, where $G(\cdot; \theta_{\text{obj}})$ is represented as a neural network with weights θ_{obj}, γ denotes the mapping to Fourier features, and $\boldsymbol{x}' = T_t(\boldsymbol{x})$ is the local coordinate representation of the (global) 2D location $\boldsymbol{x}$.

Homography to correct for non-parallel planes. Since T_t is an affine transformation, it can only model movements that are parallel to the (static) camera plane. However, in particular for the real world examples, the plane of the movements does not need to be parallel to the image plane, but could be tilted. The resulting nonlinear effects can be modeled by adding a learnable homography to the transformation from global to local coordinates. For clarity, we will not explicitly write the homography down, but rather consider it as a part of T_t. Note that no additional supervision is necessary to identify the homography.

Differentiable rendering. For rendering we evaluate the composed scene appearance at a regular pixel grid, where we use the opacity value of the local object representation to blend the color of the background and the dynamic objects. To obtain the color for the pixel $\boldsymbol{x}$, we evaluate

$$c(\boldsymbol{x}, t) = \left(1 - o(\boldsymbol{x})\right) c_{\text{bg}}(\boldsymbol{x}) + o(\boldsymbol{x}) c_{\text{obj}}(\boldsymbol{x}). \qquad (2)$$

Note that $c(\boldsymbol{x}, t)$ is time dependent due to the time dependence of the transformation T_t. This allows us to render the frames of the sequence over time.

3.3. Loss Function

We jointly optimize for the parameters of the neural implicit representations θ_{bg} and θ_{obj} and estimate the physical parameters θ_{ode} and $\mathbf{z}_0$ and the transformation parameters θ_+ and the homography matrix. To this end, we use a simple mean squared error loss between the predicted pixel values and the given images. During training we form batches of N_{batch} pixels. To make the training more stable and help the model to identify the motion of the objects, we adopt the online training approach from [57] and progressively increase the number of frames used during the optimization. More details on the training can be found in the supplement.

4. Experiments

We use four challenging physical models to evaluate our proposed approach. To analyze our method and to compare to previous work, we first consider synthetic data. Afterwards, we show that our method achieves strong results also on real-world data. For additional implementation details and results we refer the reader to the supplement.

Although several learning-based approaches that infer physical models from image data have been proposed [12, 22, 24, 59, 52], existing approaches are particularly tailored towards settings with large training corpora. However, these methods typically suffer from decreasing estimation accuracy when training data are scarce or when out-of-distribution generalization is required, as we show in the supplementary material. In contrast, our proposed approach is able to predict physical parameters from a single short video clip. Due to the lack of existing baselines tailored towards estimation from a single video, we adapt the recent work of [22] and [59] to act as baseline methods.

4.1. Synthetic Data

We compare the proposed method to two published methods [22, 59] and two baselines on synthetic data.

Two masses connected by a spring. First, we consider the two moving MNIST digits connected by an (invisible) spring on a CIFAR background, from [22], see Figure 3. The dynamics are modeled as a two dimensional two-body system. We use two separate local representations and enable the model to identify the object layering by using the maximum of both occupancy values. Besides the initial positions and velocities of the digits, the spring constant k, the equilibrium distance l are the parameters that need to be identified. To guide the model in learning which local representation represents which digit, we use an additional segmentation loss with very rough object masks as supervision on the *first* frame of the sequence. This loss is gradually reduced to enable the learning of the fine shape of the objects. For more details see the supplement.

The approach of [22] uses a learnable encoder and ve-

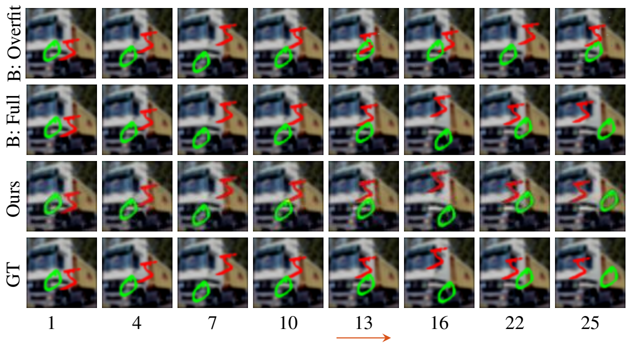

Figure 3: Two masses spring system in which MNIST digits are connected by an (invisible) spring ([22] sequence 6). The arrow indicates the start of the prediction of unseen frames. We compare our results to [22], both trained on the full dataset (B: Full) and overfitted to sequence 6 (B: Overfit). For the spring constant and equilibrium distance (k, l) the different methods achieve the relative errors $(2.7\%,\ 81.0\%)$ (B: Overfit); $(3.7\%,\ 1.8\%)$ B: Full; and $(\mathbf{0.7}\%,\ \mathbf{0.7}\%)$ (Ours). (Best viewed magnified on screen)

	PSNR ↑	Param (Mean) ↓	Param (Median) ↓
[22]: Overfit	17.66	64.77	69.55
[22]: Full	21.40	2.55	2.55
Ours	**30.30**	**2.47**	**0.76**

Table 1: PSNR and relative parameter errors ("Param") in percent for our method and the overfitted and full baseline averaged over 10 test seqs. of the MNIST digits by [22].

locity estimator to obtain initial positions and velocities of a known number of objects from the video. After integrating the known parametric model, they use a learnable coordinate-consistent decoder in combination with learned object masks and colors to render frames from the integrated trajectories. Using a photometric loss they require 5000 sequences of the same two masses spring system to train the model and identify the parameters. We report the results of their model trained on the full dataset ('B: Full'). In addition, to compare to our work in the setting of parameter estimation from a single video, we train their model on individual sequences of the test dataset ('B: Overfit').

Figure 3 shows a qualitative comparison of our results to the baseline of [22] trained in the two settings explained above. We observe, that for this sequence all approaches yield reasonable results for the reconstruction of the training frames. However, for extrapolation to unseen points in time, the overfitted model of [22] performs significantly worse, indicating that the physical model is poorly identified from a single video. While both, the baseline trained on the full dataset and our method are able to identify the parameters with high accuracy, our methods achieves an even lower error, which leads to a more precise prediction of the future frames. The fact that we achieve comparable results

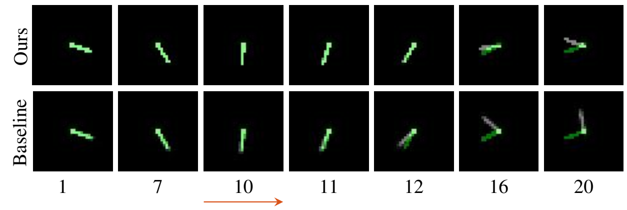

Figure 4: Prediction when training on the first 10 frames of sequence 0 of the pendulum test data by [59]. Each image shows the prediction of the respective method in white, and the ground truth as green overlay. For both methods, the prediction of images seen during training (frames 1,7,10) works well. For unseen data (frames 11,12,16,20), our method leads to more reliable predictions, meaning that our physical parameter estimation is more accurate.

while using significantly less data highlights the advantage of combining the explicit dynamics model with the implicit representation for the objects. Note that we chose sequence 6 in particular, since it yielded the best visual results for the baseline. Table 1 shows a quantitative analysis of all 10 test sequences, highlighting again the advantages of our method in the setting of a single sequence as well as the competitiveness against the usage of considerably more data. More results can be found in the supplementary material.

Nonlinear damped pendulum. We now consider the renderings of a nonlinear pendulum from [59] (cf. Figure 4). The sequences are created by OpenAI Gym [5] and contain no RGB data, but only object masks. [59] uses a coordinate aware variational encoder to obtain the initial state from object masks. After state trajectory is integrated using a learnable Lagrangian function parametrizing the dynamics of the system, a coordinate aware decoder is used to render frames from the trajectories. To compare to our method in the setting of a single video, we train the model again using only the first N frames of sequences from the test set.

In contrast to the baseline, we do not assume a known pivot point and use a more general pendulum model with damping. For a nonlinear damped pendulum the unknown parameters are the initial angle and velocity, the pivot point A, the pendulum length l and the damping coefficient c. For more details see the supplement. Since this dataset does not include image data, we employ a binary cross entropy loss wrt. the object mask using the same frames as the baseline.

Qualitative results for a single sequence are presented in Figure 4. We observe similar behavior as before: Both methods fit the given training data very well, however, in case of the baseline the pendulum motion significantly slows down for unseen time steps and the predictions for unseen data are not very accurate. We emphasize that this happens because due to the general dynamics model used, the baseline requires significantly larger training datasets, and it performs poorly in the single-video setting consid-

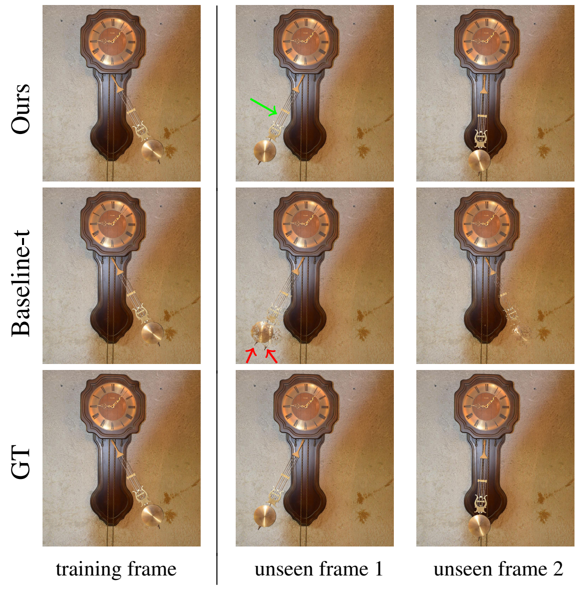

Figure 5: Rendered frames for sequence 1 of the wallclock. The left image is part of the training set, "unseen frame 1" is between two training frames, "unseen frame 2" is a future frame after the interval seen during training. While our methods makes photorealistic predictions for both unseen frames, the time-dependent background ("Baseline-t") fails in both cases. Note the visible blending between the neighboring frames in the baseline (red arrows) and the fine detail on the pendulum recovered by our method (green arrow).

ered in this paper. In contrast, our method shows a significantly better performance, which highlights the strength of directly modelling physical phenomena to constrain the learnable dynamics in an analysis-by-synthesis manner.

For a quantitative evaluation of the prediction quality, we report the intersection over union (IoU) averaged over all frames of the test sequences. Averaged over the first 20 sequences of the test set, the overfitted baseline achieves an IoU of 0.54 while our method achieves a score of 0.76. If we predict the test sequences using the baseline trained on the full dataset, we obtain an IoU of 0.73. We point out again, that our method achieves results that are en par with the baseline trained on the full dataset, while requiring only a single sequence. Moreover, as we show in the supplement, that the baseline exhibits poor generalization abilities for observations that deviate from the training distribution, while our method does not encounter such problems.

Nonlinear damped pendulum - high resolution. In contrast to the approaches of [22] and [59], we also tackle high-resolution videos with complex background and textured objects with our approach, see Figure 5. To analyze our method, we created several synthetic videos by simulating a pendulum motion with known parameters and then rendered the images of 3 different pendulums on top of each of 3 different images, creating 9 sequences per background image. The simulated sequences allow us to compare against groundtruth parameters and object masks. We select 15 frames for training and use 26 frames for evaluation. The latter frames are selected both between training frames as well as after the training interval.

To show the advantage of explicitly modelling the physical dynamics, we compare against two baselines. First, we augment the background representation by an additional input for positional-encoded time ("Baseline-t"). This gives a simple representation for a dynamic scene without any local representations. Second, we follow the idea from [57] and use a blending of background and foreground representation, where we position the foreground by learnable $\mathrm{SE}(2)$ transformations for each training frame ("Baseline-p"). To obtain time continuous transformations, we interpolate linearly between the poses estimated for the frames.

Qualitative results for a single scene can be seen in Figure 5, Table 2 shows a quantitative evaluation over all sequences. For more results we refer to the supplement. We see that our model produces photorealistic renderings of the scene, even for the predicted frames. While both baselines yield similar results on the training frames, the quality of the prediction on the test frames reduces for both methods. As can be seen in Figure 5, the time dependent background effectively blends between the training images, which means that for unseen time instances, the two pendulum positions from the neighboring training frames can be seen in the blending process. While the posed baseline does not suffer from such effects, the linear interpolation of the poses does not reflect the physical process well, and therefore the prediction quality reduces, as can be seen in Table 2. While the time dependent baseline shows undefined behavior for the prediction in the future, it is not even clear how to extrapolate the posed baseline beyond the training interval (and therefore we did not include such frames in the evaluation for this method). In contrast, our method shows physically correct prediction between the training frames and, due to the parametric physical model, is also able to make accurate predictions for future observations. We also would like to point out, that the results show, that our methods allows accurate object segmentation for the given physical systems.

4.2. Real World Data

To show the capabilities of our approach on real world data, we captured videos of three physical systems: A block sliding on an inclined plane, a thrown ball, see Figure 6, and a pendulum, see Figure 1. For the block, the initial position and velocity, the angle of the plane and the coefficient of friction are the unknown parameters. For the ball, the initial position and velocity, need to be identified. We use the model for the damped pendulum introduced earlier. See the supplementary material for the full dynamics models.

The real world data is more challenging than the syn-

	Woodwall			Stonewall			Wallclock		
	PSNR ↑	IoU ↑	Param ↓	PSNR ↑	IoU ↑	Param ↓	PSNR ↑	IoU ↑	Param ↓
Baseline-t	29.29	-	-	25.00	-	-	26.36	-	-
Baseline-p	33.48	0.86	-	31.73	0.88	-	32.82	0.86	-
Ours	**42.07**	**0.98**	0.01	**36.40**	**0.99**	0.02	**40.98**	**0.99**	0.05

Table 2: Reconstruction quality on the test frames for the synthetic examples. We report IoU of the predicted vs. groundtruth masks and the relative error of all estimated physical parameters in percent ("Param") averaged over the 9 sequences of each dataset. Our method achieves excellent reconstruction quality, mask consistency and parameter estimation, while the baselines perform worse or do not identify those quantities, which is indicated by "-".

	Pendulum		Sliding Block		Ball	
	PSNR↑	ΔH	PSNR↑	ΔH	PSNR↑	ΔH
w/o hom.	32.74	-	35.34	-	29.47	-
Full	**32.91**	0.07	**36.57**	0.18	**31.74**	0.29

Table 3: Reconstruction quality for the real world examples. The PSNR is averaged over all unseen test frames. We also show an ablation of the homography and report the Frobenius norm of the difference between the estimated homography matrix and a unit matrix (ΔH). The results show that the homography does improve the reconstruction.

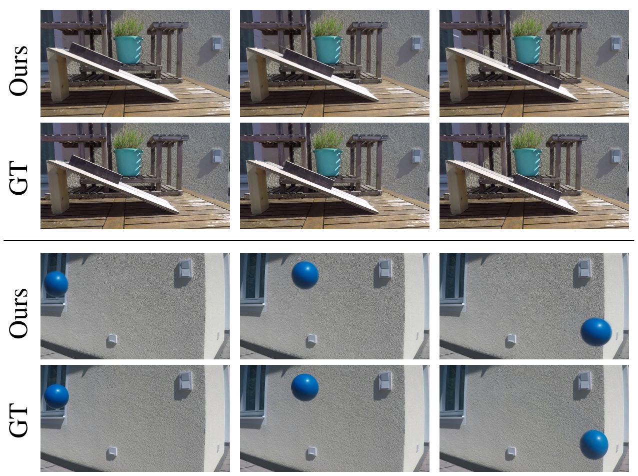

Figure 6: Reconstruction results for the sliding block and the thrown ball on test frames. Our method produces realistic predictions for previously unseen frames, confirming that the physical parameters have been identified well.

thetic data, due to image noise and motion blur. We employ the homography to account for a plane of movement that is not parallel to the image plane. For training, we extract a subset of the frames and evaluate on the remaining frames.

Table 3 shows, that we achieve very good reconstruction on previously unseen frames, which also confirms, that the physical parameters have been well identified. While groundtruth for most of the parameters is hard to acquire, the length of the pendulum, and the angle of the inclined plane are quantities that can be obtained using a ruler. The estimated quantities deviate from our measured values by 4.1% and 3.6%, respectively (relative errors). We would like to emphasize, that this shows, that for certain physical phenomena, we are able to estimate real world scale in a monocular video. To show the effectiveness of using the homography, we ablate it and report the results in Table 3.

5. Conclusion

In this work we presented a solution for identifying the parameters of a physical model from a video while also creating a photorealistic representation of the appearance of the scene objects. To this end, we proposed to combine neural implicit representations and neural ODEs in an analysis-by-synthesis fashion. Unlike existing learning-based approaches that require large training corpora, a single video clip is sufficient for our approach. In contrast to prior works that use encoder-decoder architectures specifically tailored to 2D images, we build upon neural implicit representations that have been shown to give impressive results for 3D scene reconstruction. Therefore, the extension of the proposed method to 3D is a promising direction for future work.

We present diverse experiments in which the ODE parametrizes a rigid-body transformation of the foreground objects. We emphasize that conceptually our model is not limited to rigid-body motions, and that it can directly be extended to other cases, for example to nonlinear transformations for modelling soft-body dynamics. The focus of this work is on learning a physical model of a phenomenon from a short video. Yet, the high fidelity of our model's renderings, together with the easy modifiability of the physical parameters, enables various computer graphics applications such as the artistic re-rendering of scenes, which we demonstrate in our video. Overall, our per-scene model combines a unique set of favorable properties, including the interpretability of physical parameters, the ability to perform long-term predictions, and the synthesis of high-resolution images. We believe that our work may serve as inspiration for follow-up works on physics-based machine learning using neural implicit representations.

Acknowledgements This work was supported by the ERC Advanced Grant SIMULACRON and by the DFG Forschergruppe 2987 "Learning and Simulation in Visual Computing".

References

[1] Matan Atzmon and Yaron Lipman. SAL: sign agnostic learning of shapes from raw data. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[2] Dejan Azinovic, Ricardo Martin-Brualla, Dan B. Goldman, Matthias Nießner, and Justus Thies. Neural RGB-D surface reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[3] Daniel M. Bear, Elias Wang, Damian Mrowca, Felix J. Binder, Hsiau-Yu Fish Tung, R. T. Pramod, Cameron Holdaway, Sirui Tao, Kevin A. Smith, Fan-Yun Sun, Li Fei-Fei, Nancy Kanwisher, Joshua B. Tenenbaum, Daniel L. K. Yamins, and Judith E. Fan. Physion: Evaluating physical prediction from vision in humans and machines. In *NeurIPS Datasets and Benchmarks*, 2021.

[4] Mihail Bogojeski, Leslie Vogt-Maranto, Mark E Tuckerman, Klaus-Robert Müller, and Kieron Burke. Quantum chemical accuracy from density functional approximations via machine learning. *Nature communications*, 11(1), 2020.

[5] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *CoRR*, abs/1606.01540, 2016.

[6] Pradyumna Chari, Chinmay Talegaonkar, Yunhao Ba, and Achuta Kadambi. Visual physics: Discovering physical laws from videos. *CoRR*, 2019.

[7] Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2018.

[8] Zhenfang Chen, Jiayuan Mao, Jiajun Wu, Kwan-Yee Kenneth Wong, Joshua B. Tenenbaum, and Chuang Gan. Grounding physical concepts of objects and events through dynamic visual reasoning. In *International Conference on Learning Representations (ICLR)*, 2021.

[9] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[10] Anshul Choudhary, John F. Lindner, Elliott G. Holliday, Scott T. Miller, Sudeshna Sinha, and William L. Ditto. Physics-enhanced neural networks learn order and chaos. *Phys. Rev. E*, 101:062207, 2020.

[11] Miles D. Cranmer, Sam Greydanus, Stephan Hoyer, Peter W. Battaglia, David N. Spergel, and Shirley Ho. Lagrangian neural networks. *CoRR*, abs/2003.04630, 2020.

[12] Filipe de Avila Belbute-Peres, Kevin A. Smith, Kelsey R. Allen, Josh Tenenbaum, and J. Zico Kolter. End-to-end differentiable physics for learning and control. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2018.

[13] Yilun Du, Yinan Zhang, Hong-Xing Yu, Joshua B Tenenbaum, and Jiajun Wu. Neural radiance flow for 4d view synthesis and video processing. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[14] Berthy Feng, Alexander C. Ogren, Chiara Daraio, and Katherine L. Bouman. Visual vibration tomography: Estimating interior material properties from monocular video. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[15] Chad R Galley. Classical mechanics of nonconservative systems. *Physical review letters*, 110(17), 2013.

[16] Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

[17] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *International Conference on Machine Learning (ICML)*, 2020.

[18] Vincent Le Guen and Nicolas Thome. Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[19] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[20] Raban Iten, Tony Metger, Henrik Wilming, Lídia Del Rio, and Renato Renner. Discovering physical concepts with neural networks. *Physical review letters*, 2020.

[21] Miguel Jaques, Martin Asenov, Michael Burke, and Timothy M. Hospedales. Vision-based system identification and 3d keypoint discovery using dynamics constraints. In *Learning for Dynamics and Control Conference (L4DC)*, 2022.

[22] Miguel Jaques, Michael Burke, and Timothy M. Hospedales. Physics-as-inverse-graphics: Unsupervised physical parameter estimation from video. In *International Conference on Learning Representations (ICLR)*, 2020.

[23] Navami Kairanda, Edith Tretschk, Mohamed Elgharib, Christian Theobalt, and Vladislav Golyanik. ϕ-sft: Shape-from-template with a physics-based deformation model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[24] Rama Krishna Kandukuri, Jan Achterhold, Michael Möller, and Jörg Stückler. Learning to identify physical parameters from video using differentiable physics. In *German Conference on Patter Recognition (GCPR)*, 2020.

[25] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 2021.

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

[27] Pierre Leclerc, Cédric Ray, Laurent Mahieu-Williame, Laure Alston, Carole Frindel, Pierre-Francois Brevet, David Meyronet, Jacques Guyotat, Bruno Montcel, and David Rousseau. Machine learning-based prediction of glioma margin from 5-ala induced ppix fluorescence spectroscopy. *Scientific reports*, 10(1), 2020.

[28] Nir Levine, Yinlam Chow, Rui Shu, Ang Li, Mohammad Ghavamzadeh, and Hung Bui. Prediction, consistency, curvature: Representation learning for locally-linear control. In *International Conference on Learning Representations (ICLR)*, 2020.

[29] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[30] Michael Lutter, Christian Ritter, and Jan Peters. Deep lagrangian networks: Using physics as model prior for deep learning. In *International Conference on Learning Representations (ICLR)*, 2019.

[31] Lars M. Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[32] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision (ECCV)*, 2020.

[33] Michael Niemeyer, Lars M. Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[34] Michael Niemeyer, Lars M. Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[35] Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. Neural scene graphs for dynamic scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[36] Jeong Joon Park, Peter Florence, Julian Straub, Richard A. Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[37] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[38] Songyou Peng, Michael Niemeyer, Lars M. Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *European Conference on Computer Vision (ECCV)*, 2020.

[39] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[40] Maziar Raissi and George Em Karniadakis. Hidden physics models: Machine learning of nonlinear partial differential equations. *Journal of Computational Physics*, 2018.

[41] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Numerical gaussian processes for time-dependent and nonlinear partial differential equations. *SIAM Journal on Scientific Computing*, 2018.

[42] Maziar Raissi, Paris Perdikaris, and George 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*, 2019.

[43] Maziar Raissi, Alireza Yazdani, and George Em Karniadakis. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science*, 2020.

[44] Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[45] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

[46] Liangchen Song, Sheng Liu, Celong Liu, Zhong Li, Yuqi Ding, Yi Xu, and Junsong Yuan. Learning kinematic formulas from multiple view videos. In *ACM International Conference on Multimedia*, 2021.

[47] Andrew Sosanya and Sam Greydanus. Dissipative hamiltonian neural networks: Learning dissipative and conservative dynamics separately. *CoRR*, abs/2201.10085, 2022.

[48] Karl Stelzner, Robert Peharz, and Kristian Kersting. Faster attend-infer-repeat with tractable probabilistic models. In *International Conference on Machine Learning (ICML)*, 2019.

[49] Russell Stewart and Stefano Ermon. Label-free supervision of neural networks with physics and domain knowledge. In *AAAI Conference on Artificial Intelligence*, 2017.

[50] Naoya Takeishi and Alexandros Kalousis. Physics-integrated variational autoencoders for robust and interpretable generative modeling. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[51] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[52] 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.

[53] Sebastian Weiss, Robert Maier, Daniel Cremers, Rüdiger Westermann, and Nils Thuerey. Correspondence-free material reconstruction using sparse surface constraints. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[54] Jiajun Wu, Ilker Yildirim, Joseph J. Lim, Bill Freeman, and Joshua B. Tenenbaum. Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2015.

[55] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[56] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Ronen Basri, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[57] Wentao Yuan, Zhaoyang Lv, Tanner Schmidt, and Steven Lovegrove. Star: Self-supervised tracking and reconstruction of rigid objects in motion with neural rendering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[58] Yaofeng Desmond Zhong, Biswadip Dey, and Amit Chakraborty. Symplectic ode-net: Learning hamiltonian dynamics with control. In *International Conference on Learning Representations (ICLR)*, 2020.

[59] Yaofeng Desmond Zhong and Naomi Ehrich Leonard. Unsupervised learning of lagrangian dynamics from images for prediction and control. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.