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

Anonymous Author(s)
Affiliation
Address
email

Abstract

Extracting physical dynamical system parameters from videos, which is important for scientific and technological applications. Current methods rely on supervised deep networks trained on large labeled datasets which are difficult to acquire. Existing unsupervised techniques, which rely on frame prediction, have limitations such as long training times, instability, and applicability to specific motion problems. The proposed method addresses these issues by estimating physical parameters of any known, continuous governing equation from single videos using a KL-divergence-based loss function in the latent space. This approach is robust, works for various systems beyond motion, and eliminates the need for frame prediction, reducing model size and computation.

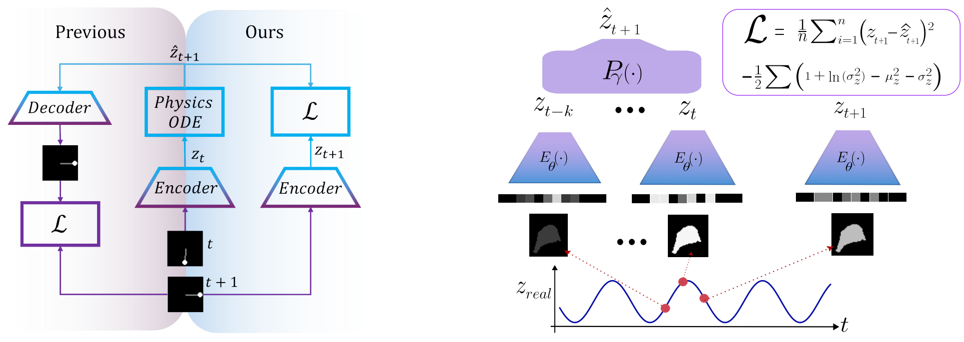
1 Introduction

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 [6, 18, 32]. Use-cases include trajectory prediction for celestial objects [15], healthy and diseased tissue characterization [14], and physical model validation [6, 7]. Fitting governing equations [2] often requires using additional sensors to directly measure system states. Instead, doing measurements from a video allows to avoid additional sensors, yet, requires manually labelling pixels or video frames which is time-consuming and expensive. Therefore, automated and unsupervised methods are needed to extract dynamics from videos and accurately estimate physical parameters [6, 15, 18, 17, 32].

Recent work addressed parameter estimation from video by deep learning [6, 7, 37] or reinforcement learning [3]. For instance, proposed supervised learning methods rely on datasets with extensive and high precision labels which are exceedingly difficult to obtain [1, 4, 26, 28, 35]. To avoid labeling, current unsupervised methods for estimating physical parameters build on encoder-decoder network designs: reconstructing video frames from low-dimensional representations. The frame reconstruction is a mere by-product of the paramter estimation, and leads to overly complex solutions which are difficult to train [15, 18]. Along with this, current solutions [15, 18, 20, 38, 43] are constrained to motion dynamics, excluding a wide variety of systems like dynamics related to brightness, colour, and deformations, among others [15, 18].

Our work proposes an unsupervised learning method to solve the inverse problem using videos of a dynamical system with known, continuous governing physics equations. Our method can be implemented for different dynamical systems beyond motion. Unlike previous approaches, we present an evaluation of the latent space of our model in multiple systems. We show that our unsupervised model fits the dynamics and effectively generalizes to unseen future frames. In addition, we bypass frame reconstruction by calculating the loss in the latent space, eliminating the need for reconstruction. Our approach is thus fast, less resource-intensive and more robust to initial conditions compared to existing methods.

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.



(a) (b)

Figure 1: We propose a novel unsupervised approach to physical parameter estimation from videos. **(a)** Starting from a video frame (e.g. of a pendulum) at time t, parameter estimation techniques encode the dynamical states z_t. A Physics ODE with learnable parameters solves the governing equation to predict future states $\hat{z}_{t+1}$ in the latent space (blue lines). Previous methods design decoders to reconstruct a frame at time $t+1$ and use reconstruction loss ($\mathcal{L}$, purple) to train the Physics ODE. In contrast, our method avoids the decoder by leveraging a loss function in the latent space ($\mathcal{L}$, blue). **(b)** A video (bottom) displays intensity dynamics z_{real}. Frames are encoded by $E_\theta(\cdot)$ to latent representations z_t. The physics block $P_\gamma(\cdot)$ predicts the future step $\hat{z}_{t+1}$, which is compared to z_{t+1} encoded from the corresponding frame. **Top-right:** Proposed loss function; the first term ensures the prediction fits with the encoding, the second expression controls z variance.

2 Related Work

The relationship between physics and deep learning (DL) is symbiotic. Physics inspired segmentation [25], generative models [34, 33] and new architectures [12]. Likewise, DL is used to study, understand and create new physics from data [6, 7, 17, 37]. Techniques like physics-informed neural networks (PINN) [21, 31] or Lagrangian neural networks (LLN) [8, 27] are designed to solve inverse problems. Yet, PINNs are constrained to initial/boundary conditions and time reference [21, 11, 29]. Moreover, these methods are supervised and require labeled data which can be expensive or infeasible to obtain [1]. Therefore, our work focuses on the inverse problem from video using unsupervised machine learning. Some works on learning physics from videos focus on frame prediction [5, 10, 13, 24] but not on parameter estimation. Besides, research on extracting physical information from video requires labelled datasets, dynamical variables or parameters' ground truth [40, 9, 27, 31, 39, 37, 42, 41]. Moreover, these methods focus on motion problems and based on interaction networks [5, 37, 36] and some aims to parameterize deformations [38, 19].

Some works with **unsupervised parameter estimation from video** use frame representation and physics priors of known governing equations with unknown parameters [38, 18, 15, 43, 20]. Some approaches [20, 43] use variational auto-encoders (VAE) [23] with a physics engine; however, the reconstructions are poor and constrained to motion problems. Works similar to ours [15, 18] estimate parameters from a single video without annotations. Our method compares favourably in terms of robustness to initializations, latent space interpretability and it is not constrained to motion. **Baselines:** Jaques et al. [18] uses an auto-encoder with a physics engine to reconstruct inputs and generate future frame predictions. For objects of interest, a U-Net model learns segmentation masks, which are necessary for the spatial transformer (ST) they use in the decoder [16]. The spatial transformer performs an affine transformations on the mask to 'move' the object based one predictions. Second, Hofherr et al. [15] uses a differentiable ODE solver to estimate the parameters. This model also needs a ST, but at the pixel level, the pixels are displaced using the prediction made by the ODE solver. The model needs to be trained with masks to learn which pixels should be translated. However, using frame reconstruction to achieve parameter estimation is challenging and makes the network slow to train since reconstructing frames from low dimensional data (i.e., a set of positions) is an ill-defined problem. Therefore, [15, 18, 38] limited their scope by using a mask and a spatial transformer [16], excluding dynamical systems with changes in intensity/colour, deformations and non-uniform scaling among others which we explicitly allow in our paper.

3 Methods

Our approach estimates the parameters of a known governing equation from a video with unannotated frames and known frame rate δt. We use a simple encoder and a physics block. Figure 1b shows our approach. This work is scoped to dynamics given by autonomous differential equations (Eq.1a), which depend on the state variable captured in the video.

$$\textbf{a)} \quad z^{(n)} + \gamma_{n-1} z^{(n-1)} + ... + \gamma_1 z^{(1)} + \gamma_0 z = 0, \quad \textbf{b)} \quad z^{(2)} + \gamma_1 z^{(1)} + \gamma_0 z = 0. \tag{1}$$

This is an n^{th}-order system, where z is the time-dependent state variable, $z^{(k)}$ $k = 1, 2, \ldots n$ is the k^{th}-derivative of z with respect to time t and γ_i, $i = 0, 1, ..n - 1$ are the parameters of the equation to estimate. As a *proof-of-concept* we first consider a second-order differential equation (Eq. 1b).

The **encoder** is a neural network $E_\theta(x)$ that maps images $x \in \mathbb{R}^{w \times h \times c}$ to the state variable $z \in \mathbb{R}^d$. The **physics block** numerically solves the differential equation using Euler's method, where γ_i are learnable parameters and the predicted latent space $\hat{\mathbf{z}}_{t+1}$ is a function $P_\gamma(\cdot)$ of the latent representations for the n previous frames. The first part $\mathcal{L}_1$ of our **loss function** minimizes the difference between $\mathbf{z}$ and $\hat{\mathbf{z}}$. However, convergence to trivial solutions like $E_\theta(x) = 0\,\forall x$ and $P_\gamma(z) = 0\,\forall z$ poses a problem. To avoid it, we propose to induce variance in the encoder's output assuming latent variables $z_k \in \mathcal{N}(\mu, \sigma^2)$ as a re-normalization of the metric. The second part of the loss function $\mathcal{L}_2$ uses the Kullback-Leibler divergence (KL-divergence). Thus, z_k is an element of the random variable $Z \sim \mathcal{N}(\mu_z, \sigma_z^2)$ and we want it to follow a particular distribution $Q \sim \mathcal{N}(0, 1)$.

$$\mathcal{L}_1 = \frac{1}{n} \sum_{i=1}^{n} (z_i - \hat{z}_i)^2, \quad \mathcal{L}_2 = \mathrm{KL}(Z||Q) = -\frac{1}{2} \sum_{i=1}^{D} \big(1 + \ln(\sigma_z^2) - \mu_z^2 - \sigma_z^2\big) \tag{2}$$

We use KL-divergence differently than conventional VAEs [23] which 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 μ_z and variance σ_z^2 over the batch. Finally, our loss function is given by $\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2$.

4 Experiments

We evaluate our model on three different continuous dynamical systems involving motion, intensity and scaling. Figure 2a illustrates our three simulated datasets, which are comparable to datasets used to evaluate state-of-the-art methods [9, 18, 36, 37, 41, 42]. Dataset details are in Appendix A.2.

4.1 Latent space evaluation and parameter estimation

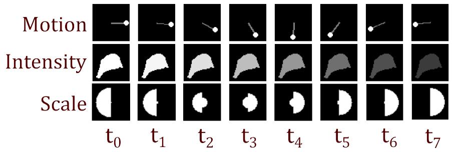
We train our model using the three synthetic datasets and evaluate the dynamics z estimated by the encoder $E_\theta(\cdot)$. We compare the estimated state variable z to its ground truth value z_{real}, which was used to generate the data. Following Eq. 1b, we consider second-order dynamics for the three datasets, where the evolution of the state variable z follows a dampened oscillation.

Figure 2 shows that the model is capable of estimating the dynamics z for all three datasets. Although oscillatory dynamics can be challenging for neural networks, we find that our unsupervised loss fits the dynamical behaviour with small deviations from the ground truth. Figure 2d depicts the encoder output z against the 'ground truth' z_{real} for every frame the train and test set (dots). Importantly, our model output has physical interpretability since we directly use the differentiable ODE (Eq. 1b) during training. Due to these physics priors, the model is able to generalize at test time to inputs unseen during training. The network's extrapolation of z to unseen future time steps is shown using dashed lines in Figure 2a.

Table 2b shows the model's accuracy to estimate γ, where γ_0 is the oscillation frequency and γ_1 is the damping factor. We observe that the most accurate results are obtained using the 'Intensity' dataset because the temporal information is in the pixel intensities instead of location, which means, information about the dynamics is less discretized than the other datasets. Specifically, using 8-bit integer values for pixel intensities, the input can assume 256 different values. In contrast, with motion and scaling, the dynamics are discretized by the pixel locations. In particular, for a (50×50) frame size, the discretization of the dynamic variable is more impactful, especially as the oscillation amplitude decreases. This effect is seen in the dynamics of the 'Scale' dataset in Figure 2a (red line). besides the inaccuracy in estimating parameter $\gamma_1$in the motion and scale dataset is anticipated from the discretization discussed.

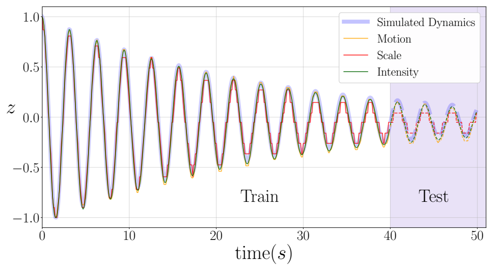
4.2 Real-world video evaluation

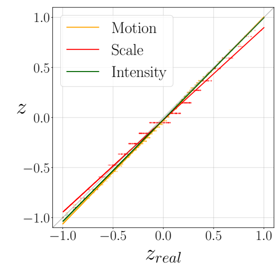
Figure 3 presents an evaluation on a real-world video dataset we recorded, where the objective is to estimate the rope length which cannot be seen in the video, but is known to be 120 cm. It can be seen the approach is accurate with an error of 2cm, showing the capabilities of the method in extracting information not explicitly shown in the video: in the same way, we analyzed the latent space showing the natural behaviour of a pendulum with its expected natural damping.



	Motion	Intensity	Scaling	Expected
γ_0	3.943 ± 0.008	3.887 ± 0.035	4.055 ± 0.026	4.00
γ_1	0.144 ± 0.007	0.0889 ± 0.010	0.910 ± 0.012	0.08

(a) (b)

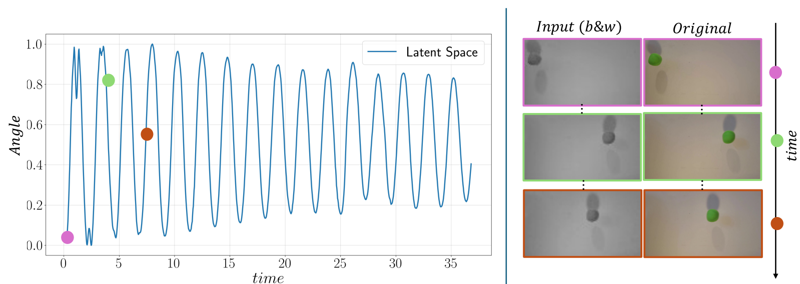




(c) (d)

Figure 2: **(a)** Example frames from our datasets. Each row shows a different dataset, corresponding to a different dynamical system, and each column a different time sample. **(b)** Parameter estimation accuracy: Mean and standard deviation of each learnable parameter in the physics block after training. Rightmost column shows the ground truth values. **(c)** Latent space estimation of the dynamic variable z for the three datasets. The blue line shows the 'ground truth' value z_{real} of the simulated dynamics, the model was trained with the dynamics of the continuous line while the dotted line is the predictions of the test set. **(d)** Encoder output z vs. z_{real} for every frame.

Equation	**Parameter**	**Expected** (m)	**Estimated** (m)
$\theta^{(2)} = -\zeta\theta^{(1)} - \frac{g}{L}\sin(\theta)$	String length (L)	1.20	1.22



(b) (c)

Figure 3: Real-world pendulum recording parameter estimation. **(a)** The angle θ is the latent variable. Damping factor ζ and the string length L are learned. **(b)** Extracted dynamics by the model. **(c)** Gray scale input and the original frame from the dataset, related to time in plot (b) using the coloured dots. Our model can estimate the parameter L with only a 0.02 m error.

5 Discussion and Limitations

We present a novel method for physical parameter estimation of governing equations. While previous methods in literature do not study phenomena other than motion, we go beyond motion and include a variety of dynamical systems while avoiding frame prediction. We examine latent space predictions directly, unlike state-of-the-art models [15, 18, 20] which do not discuss the accuracy of predicted dynamics, but only report frame reconstruction accuracy. However, when the objective is parameter estimation, reconstruction is simply a tool to define the unsupervised loss, and the latent space should be analysed closely. Our model does not resolve the absolute scale of the state variable. Yet, thanks to the loss function, the model is normalized, ensuring assumptions made in section 3. Baselines implicitly or explicitly do this normalization using the spatial transform, forcing the prediction to be in the pixel metric. Finally, our successful parameter estimation without masks on a real-world video is an improvement over baselines.

Limitations We used continuous, autonomous differential equations. However, some systems such as fluids are described with more complex differential equations. In addition, we need to guarantee that the model can be differentiated. Our proposed model remains to be explored with an extension of the experiments and more complex use-cases, which might include combining dynamics or multiple entities in a scene with independent dynamics.

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*, volume 664, page 072015. IOP Publishing, 2015.

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

[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.

[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.

[5] 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.

[6] 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.

[7] 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.

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

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

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

[19] 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.

[20] 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.

[21] 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.

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[23] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[24] Jannik Kossen, Karl Stelzner, Marcel Hussing, Claas Voelcker, and Kristian Kersting. Structured object-aware physics prediction for video modeling and planning. *arXiv preprint arXiv:1910.02425*, 2019.

[25] 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.

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

[27] Michael Lutter, Christian Ritter, and Jan Peters. Deep lagrangian networks: Using physics as model prior for deep learning. *arXiv preprint arXiv:1907.04490*, 2019.

[28] 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.

[29] 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.

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

[31] 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. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2018.10.045. URL `https://www.sciencedirect.com/science/article/pii/S0021999118307125`.

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

[33] 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.

[34] Peter Toth, Danilo Jimenez Rezende, Andrew Jaegle, Sébastien Racanière, Aleksandar Botev, and Irina Higgins. Hamiltonian generative networks. *arXiv preprint arXiv:1909.13789*, 2019.

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

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] Jiajun Wu, Erika Lu, Pushmeet Kohli, Bill Freeman, and Josh Tenenbaum. Learning to see physics via visual de-animation. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper_files/paper/2017/file/4c56ff4ce4aaf9573aa5dff913df997a-Paper.pdf`.

[41] 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.

[42] David Zheng, Vinson Luo, Jiajun Wu, and Joshua B Tenenbaum. Unsupervised learning of latent physical properties using perception-prediction networks. *arXiv preprint arXiv:1807.09244*, 2018.

[43] Yaofeng Desmond Zhong and Naomi Leonard. Unsupervised learning of lagrangian dynamics from images for prediction and control. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 10741–10752. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/79f56e5e3e0e999b3c139f225838d41f-Paper.pdf`.

A Appendix / supplemental material

A.1 Equations

Encoder

$$E : \mathbb{R}^{w \times h \times c} \to \mathbb{R}^d, \quad z_t = E_\theta(x_t). \tag{3}$$

Eulers Method

$$z_t^{(1)} \approx \frac{z_{t+1} - z_t}{\delta t} \approx \frac{z_t - z_{t-1}}{\delta t}, \quad z_{t+1} = z_t + \delta t z_t^{(1)}, \quad z_{t+1}^{(1)} = z_t^{(1)} + \delta t z_t^{(2)}. \tag{4}$$

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.

$$\mathbf{z} = \begin{bmatrix} z_n \\ \vdots \\ z_{t+1} \\ \vdots \\ z_T \end{bmatrix} = \begin{bmatrix} E_\theta(x_n) \\ \vdots \\ E_\theta(x_{t+1}) \\ \vdots \\ E_\theta(z_T) \end{bmatrix} \tag{5}$$

$$\hat{\mathbf{z}} = \begin{bmatrix} \hat{z}_n \\ \vdots \\ \hat{z}_{t+1} \\ \vdots \\ \hat{z}_T \end{bmatrix} = \begin{bmatrix} P_\gamma(z_{n-1}, \ldots, z_0; \gamma) \\ \vdots \\ P_\gamma(z_t, \ldots, z_{t-n}; \gamma) \\ \vdots \\ P_\gamma(z_{T-1}, \ldots, z_{T-n}; \gamma) \end{bmatrix} \tag{6}$$

Extension of Loss $\mathcal{L}_1$

$$\mathcal{L}_1 = \frac{1}{n} \sum_{i=1}^{n} (z_i - \hat{z}_i)^2 = \frac{1}{n} \sum_{i=1}^{n} \Big[E_\theta(x_i) - P_\gamma\big(E_\theta(x_{i-1}), \ldots, E_\theta(x_{i-n})\big) \Big]^2 \tag{7}$$

A.2 Dataset Specifications

Motion systems can be understood as problems where pixels have a translation or rotation transformation; classic examples include pendulums, springs, and celestial movements. For the motion system, we simulated a pendulum as the majority of related literature uses this example where the state variable is the angle of the pendulum.

Intensity consists of the change in grayscale pixel values. This problem can be seen in nature in voltage imaging of neurons or photonic crystals. For this dataset, we generated an shape and assigned inside pixels the corresponding value of dynamical system.

Scale in a video means changes in the total number of pixels corresponding to our object of interest. It can be related to a growing bacteria population or liquid diffusion. For this dataset, we created a filled circle centred 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.

Each uses an image size of (50×50) pixels, and the simulated dynamics are the same for the three datasets (Eq. 1b) normalized with respect to the image size and maximum pixel intensity. Each dataset consists of 500 training samples with 20 frames, with a final numerical dimensionality of $(samples \times frames \times \#channels \times width \times height) = (500 \times 20 \times 1 \times 50 \times 50)$.

A.3 Network specification

The encoder consists of three linear layers with ReLU activation functions in the first and second layer.

A.4 Training

The experiments and baselines were executed on a GPU NVIDIA 3080. For our model, implemented in PyTorch, the encoder was trained using the Adam optimizer [22]with a learning rate of $1 \times e^{-2}$ and the default weight initialization for MLP layers.

Training: For parameter estimation in the physics block $P_\gamma(\cdot)$, we used a learning rate proportional to the initial value γ_k^0 of the learnable parameter γ_k, where $\mathrm{lr}_{(\gamma_k)} \sim 10^{[\log_{10}|(\gamma_k^0)|]}$. This approach provides sufficiently large step sizes at the beginning of training to escape of local minima. Further training details are in Appendix A.4.

A.5 Simulation Details

Here we discuss the details of the dynamics simulation of the syntetic experiments. The equation 1b represents and harmonic oscillator with close solution:

$$z(t) = Ae^{-\zeta t} cos(\omega t + \phi) \tag{8}$$

Where $\omega = 2$ is the frequency we used for simulation and $\zeta = 0.04$ the damping factor. this parameter relates to γ as follows:

$$\gamma_0 = \omega^2 + \zeta^2 = 4.0016 \tag{9}$$
$$\gamma_1 = 2\zeta = 0.08 \tag{10}$$

A.6 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. In fact, it is known that the models presented in baselines are sensitive to initialization and may fail to converge [15, 30]. In Figure 4, we evaluate the robustness of our model against changes in parameter initializations. We initialize the learnable parameters γ of Eq. 1b in the interval $[-10.0, 10.0]$ over multiple runs. The ground truth values used to generate the synthetic datasets were $\gamma_0 = 4$ and $\gamma = 0.08$. We show the convergence of the parameter prediction during training with different initializations. As can be seen, the model converges close to the ground truth values for each dataset experiment.

A.7 Baseline comparison

The dataset used to compare the baselines, both baselines were tested in the dataset published by [18] and also used in [15], the equations of motion used for each system are presented in Eq. 11

$$\vec{F}_{ij} = -k(\vec{p}_i - \vec{p}_j) - l\frac{\vec{p}_i - \vec{p}_j}{|\vec{p}_i - \vec{p}_j|} \tag{11}$$

We did not use our datasets in the comparison since baselines are not designed to handle our intensity and scale setting. Therefore, for a fair comparison, we use the dataset first proposed in [18] and reused in [15].

	Number parameters	**Time epoch** (s)	**Uses Decoder**	**Inputs Masks**
PAIG [18]	5.27M	252.72	✓	✗
PAIG w/o U-Net	4.78M	80.56	✓	✓
NIRPI [15]	75.42K	0.11	✓	✓
Ours	4.19M	0.95	✗	✓

Table 1: Relevant differences between our model and the baselines.

In Table. 1, we present a size comparison of the models along with the training time for one epoch when using the setup proposed by the authors. PAIG does not require object masks as input since it learns the segmentation via a U-Net in its pipeline.

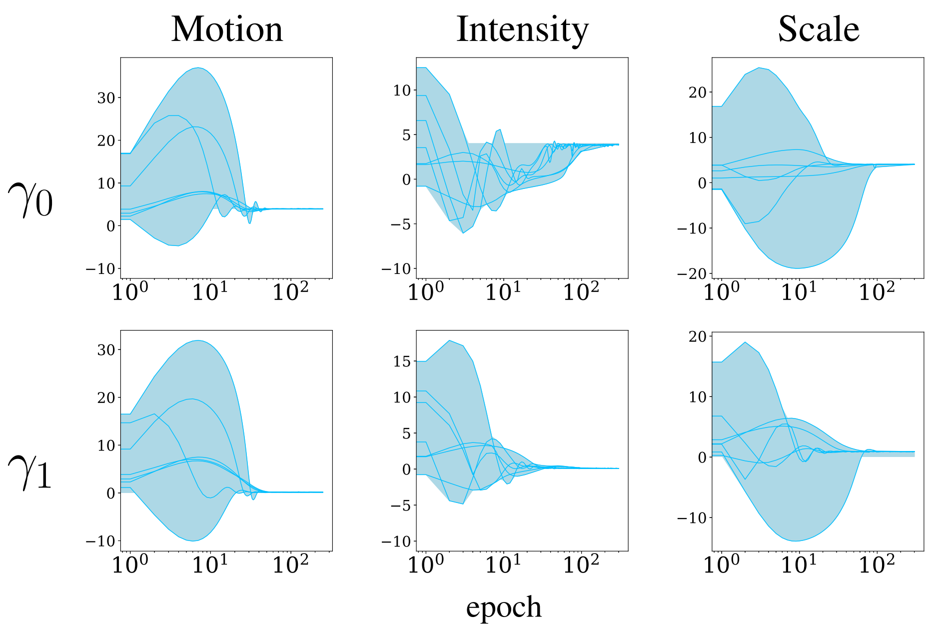


Figure 4: Robustness of the parameters estimation against different initializations. The columns indicate the different dynamical systems, while the rows are the parameters to estimate. Blue lines show the value of the estimated parameter γ_i over training epochs. Since convergence was relatively fast, the horizontal axis is on a logarithmic scale for visibility. The shading highlights the variance of the trajectories before convergence.

Figure 5: **Dataset baseline**. It shows the evolution of the spring dynamical system of two MNIST digits over a static CIFAR10 background. Figure edited from [18]

In addition, we empirically demonstrate the baseline's sensitivity to initial conditions: We train each model twice in the experiment and initialize the estimated parameter k with values 1.0 and 10.0, respectively. The expected value is $k = 2$ as used in the baseline papers [18, 15]. In Figure 6, 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. Besides, this experiment shows our model is suitable for two dimensional problems with multiple objects.

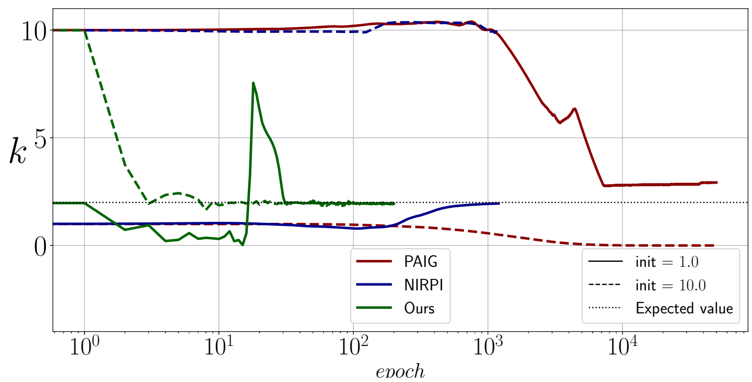


Figure 6: Robustness of the parameter estimation compared to baselines. For a fair comparison, we use the synthetic dataset created originally by the authors of the baseline papers to evaluate their models (see Appendix A.7). We plot the trajectories of the estimated parameter k during training with 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 $k = 2.0$.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: All justifications have their corresponding experiment or explanation, and the paper tries to follow the claim order of the introduction and abstract

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: Yes a subsection in the Discussion and Limitations sections addresses all the concerns we encounter in the development of the project.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: whether the assumptions about choices are fully explained in the document, we didn't make any theoretical proofs.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: **[TODO]**

Justification: To be reproducible, we explained all experiments, data-base generation and setup in detail. Besides, the code and data will be available.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code and data will be available.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: This is done in detail on the methodology section

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: the experiments present mean and standard deviation, besides one of the cores of our paper consist in the comparison of multiple realisations

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: This part is detailed in the methods, also in the results compared against baselines.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The code of ethics was read and followed

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: Our paper presents a method to study a dynamical system in a particular setting under specific conditions. Although these methods are designed for science, we stated that further research needs to be still applied in order to implement this method in real data, and we can foresee possible harm or negative uses of it

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [Yes]

Justification: Our project is really fundamental in an area not well explored , therefore we do not consider it to be risk

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [NA]

Justification: Yes, one database was used from the original authors and it is well referenced

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: So far we have not liberated or publish new assets and will be consider uopn acceptance

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: No human participants were used in this project

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: No human participants were used in this project

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.