VID2PARAM: ONLINE SYSTEM IDENTIFICATION FROM VIDEO FOR ROBOTICS APPLICATIONS

A PREPRINT

Martin Asenov
School of Informatics
The University of Edinburgh
m.asenov@ed.ac.uk

Michael Burke
School of Informatics
The University of Edinburgh
michael.burke@ed.ac.uk

Daniel Angelov
School of Informatics
The University of Edinburgh
d.angelov@ed.ac.uk

Todor Davchev
School of Informatics
The University of Edinburgh
t.b.davchev@ed.ac.uk

Kartic Subr
School of Informatics
The University of Edinburgh
K.Subr@ed.ac.edu

Subramanian Ramamoorthy
School of Informatics
The University of Edinburgh
s.ramamoorthy.ed.ac.uk

July 15, 2019

ABSTRACT

Robots performing tasks in dynamic environments would benefit greatly from understanding the underlying environment motion, in order to make future predictions and to synthesize effective control policies that use this inductive bias. *Online* system identification is therefore a fundamental requirement for robust autonomous agents. When the dynamics involves multiple modes (due to contacts or interactions between objects), and when system identification must proceed directly from a rich sensory stream such as video, then traditional methods for system identification may not be well suited. We propose an approach wherein fast parameter estimation with a model can be seamlessly combined with a recurrent variational autoencoder. Our Physics-based recurrent variational autoencoder model includes an additional loss that enforces conformity with the structure of a physically based dynamics model. This enables the resulting model to encode parameters such as position, velocity, restitution, air drag and other physical properties of the system. The model can be trained entirely in simulation, in an end-to-end manner with domain randomization, to perform online system identification, and probabilistic forward predictions of parameters of interest. We benchmark against existing system identification methods and demonstrate that Vid2Param outperforms the baselines in terms of speed and accuracy of identification, and also provides uncertainty quantification in the form of a distribution over future trajectories. Furthermore, we illustrate the utility of this in physical experiments wherein a PR2 robot with velocity constrained arm must intercept a bouncing ball, by estimating the physical parameters of this ball directly from the video trace after the ball is released.

***K*eywords** Model Calibration, Predictive Model, Variational Recurrent Neural Networks

1 Introduction

Robots that must adapt to a dynamic environment need to reason about the dynamics in this environment. Traditionally, the complexity of this reasoning has been avoided by investing in fast actuators and very accurate sensing (e.g., motion capture equipment in the lab). In emerging field applications of robotics, where the reliance on such infrastructure may need to be decreased, there is a need for better physical scene understanding, and the ability to make forward predictions of the scene.

Techniques for system identification, originally developed for process control domains, are directly aimed at this problem. There are a number of different approaches to estimating parameters, and sometimes even model structure, from observed data. Typically, these methods are best suited to domains where learning can be performed in batch mode, and also the mapping from rich sensory streams to the reduced model state space can be handled by other modules (e.g., with tools for object detection, tracking and so on).

We focus on the case where the robot must perform robust system identification *online*, directly from a rich sensory stream such as video (including the implicit tasks of detecting and tracking objects). Furthermore, we would like to structure the learned model so as to be able to make probabilistic future predictions, to enable appropriate action selection.

We present a model that extends the Variational Recurrent Neural Network (VRNN) [13]. We add an additional loss term for encoder-decoder mapping from a given sensory input (vision) to physical parameters and states (position, velocity, restitution factor, gravity, etc. in a parametric description of a physics-based model). We show that such a model can be trained with suitable domain randomization in simulation, and deployed in a real physical system. Such a model, which we call Vid2Param, allows for forward predictions envisioning possible future trajectories based on uncertainty in the estimate of the physical parameters. To illustrate the utility of this capability, we demonstrate this model on the task of intercepting a bouncing ball with a relatively slow moving robot arm and standard visual sensing.

2 Related Work

2.1 System Identification

System identification (SysID) is concerned with the problem of determining the structure and parameters of a dynamical system, for subsequent use in controller design. The best developed versions of system identification methods focus on the case of linear time-invariant (LTI) systems, although of course all of these methods have also been extended to the case of nonlinear and hybrid dynamical systems. With these more complex model structures, the computational complexity of identification can be relatively high even for moderately sized data sets.

Examples of system identification procedures that could be applied to our problem domain, including the additional step of reducing model order, include the Eigen system realization algorithm [24] and Balanced POD (BPOD) [36] (which theoretically obtain the same reduced models [27]), and the use of feedforward neural networks [11]. BPOD can be viewed as an approximation to another popular method, Balanced truncation (BT) [37], which scales to larger systems.

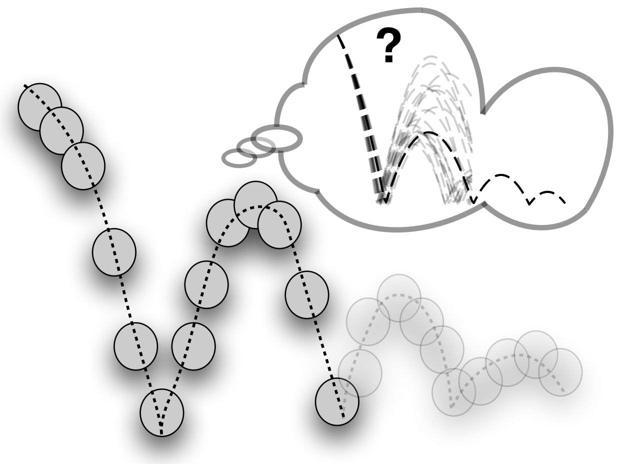


Figure 1: **Inference and generation.**

Another way to approach the problem of identification is frequency domain decomposition [6] [7]. Recent approaches in this vein include DMD [26] and Sindy [8], which allow for data driven, model-free system identification and can scale to high-dimensional data. When performing SysID directly from a rich sensory stream like video, it is not always clear what the optimal reduced representation should be [1]. We exploit the fact that a physics based model of objects can provide useful regularisation to an otherwise ill-posed identification problem.

2.2 Simulation alignment

When a parametric system model is available, simulation alignment can be performed to identify the parameters of the system. A standard approach is to perform least squares minimization, for instance computing best fit parameters to align simulator traces to observed data [30]. When simulation calls are expensive, a prior over the parameter space can be enforced, e.g. Gaussian Processes, and a Bayesian Optimization can be used [35] [32] [34]. Our approach is closely related to [43] as we use supervision during the training phase of our model, and then use this learned approximation at test time. We also employ domain randomization while training our model [31] and our work follows a similar line of reasoning to that of [10], which aims to align a simulator to real world observations as the model is being trained. We focus on the problem of aligning a model to online observations at test time, for predictive purposes.

2.3 Learnable Physics Engines

There has been increasing interest in learnable physics engines - for example learning complex factorization (at the object or particle level) from data [9] [28] [5], using particle based networks for fluid dynamics [38] and in robotics [14] [15]. By representing the problem in terms of a set of learnable components (graph representing objects/particle and relations, Navier Stokes equations, linear complementary problem for the above mentioned tasks) a physics engine can be learned from raw data. Similar approaches have been shown to scale to video data [40]. We explore the complementary problem of system identification (with an analytical or learned simulator), and propose a direct optimization approach by learning an inverse probabilistic physics engine. This builds upon ideas presented in [41], where an analytical simulator is used with traditional system identification approaches. Closely related work is presented in [33], where surface properties are learned using Physics and Visual Inference Modules.

A related question to learning interactions between objects is that of learning a state space system to represent these. This has been explored for individual objects [25] [17], by using Kalman and Bayes filters for learning. State models and predictions have recently been explored in the context of videos involving multiple objects [20] through the use of Spatial Transformer Networks [21] and decompositional structures for the dynamics, as well as integrating the differential equations directly into a network [22].

2.4 Variational Autoencoder

Variational Autoencoders (VAEs) have been extensively applied to image and video generation [16] [20] [23]. Recently, VAEs have been used in reinforcement learning to improve generalization by learning a master policy from a set of similar MDPs [4]. Closely related work is that of [2] where Variational RNNs are used to learn the 'residual physics' [42] [39]. The addition of loss terms to the reconstruction and KL error terms have also been proposed, allowing for enforcement of multiple desired constraints [19] [3]. We extend this line of work, by demonstrating that such constraints can be applied in a recurrent model to satisfy physics properties.

3 Background: Variational Recurrent Neural Networks

Our model is based on the recurrent VAE (VRNN) for modelling sequences [13]. The network consists of an RNN of the form $\mathbf{h}_t = f_\theta\left(\mathbf{x}_t, \mathbf{h}_{t-1}\right)$ with a VAE providing the hidden state h_{t-1} at each time step, in addition to the input x_t:

$$\begin{aligned}\mathbf{z}_t|\mathbf{x}_t \sim \mathcal{N}\left(\boldsymbol{\mu}_{enc,t}, diag\left(\boldsymbol{\sigma}^2_{enc,t}\right)\right), \text{ where } \boldsymbol{\mu}_{enc,t}, \boldsymbol{\sigma}_{enc,t} &= \varphi_\tau^{enc}\left(\varphi_\tau^{\mathbf{x}}\left(\mathbf{x}_t\right), \mathbf{h}_{t-1}\right) \\ &= q\left(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t}\right)\end{aligned} \tag{1}$$

Additionally, the distribution is no longer conditioned on a normal distribution $\mathcal{N}\left(0,1\right)$, but on the hidden state h_{t-1}:

$$\begin{aligned}\mathbf{z}_t \sim \mathcal{N}\left(\boldsymbol{\mu}_{prior,t}, diag\left(\boldsymbol{\sigma}^2_{prior,t}\right)\right), \text{ where } \boldsymbol{\mu}_{prior,t}, \boldsymbol{\sigma}_{prior,t} &= \varphi_\tau^{prior}\left(\mathbf{h}_{t-1}\right) = \\ &= p\left(\mathbf{z}_t|\mathbf{x}_{<t}, \mathbf{z}_{<t}\right)\end{aligned} \tag{2}$$

Similarly, the generation distribution is expanded in comparison to a standard VAE, by including the hidden state h_{t-1}:

$$\mathbf{x}_t|\mathbf{z}_t = \varphi_\tau^{dec}\left(\varphi_\tau^{\mathbf{z}}\left(\mathbf{z}_t\right), \mathbf{h}_{t-1}\right) = p\left(\mathbf{x}_t|\mathbf{z}_{\leq t}, \mathbf{x}_{\leq t}\right) \tag{3}$$

Finally, the recurrent step is implemented by including x_t and z_t in addition to h_{t-1}:

$$\mathbf{h}_t = f_\theta\left(\varphi_\tau^{\mathbf{x}}\left(\mathbf{x}_t\right), \varphi_\tau^{\mathbf{z}}\left(\mathbf{z}_t\right), \mathbf{h}_{t-1}\right) \tag{4}$$

Thus the overall loss, with the KL term and reconstruction loss, becomes:

$$\mathbb{E}_{q\left(\mathbf{z}_{\leq T}|\mathbf{x}_{\leq T}\right)}\left[\sum_{t=1}^{T}\left(-\mathrm{KL}\left(q\left(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t}\right)\|p\left(\mathbf{z}_t|\mathbf{x}_{<t}, \mathbf{z}_{<t}\right)\right) + \log p\left(\mathbf{x}_t|\mathbf{z}_{\leq t}, \mathbf{x}_{<t}\right)\right)^2\right] \tag{5}$$

.

4 Model overview

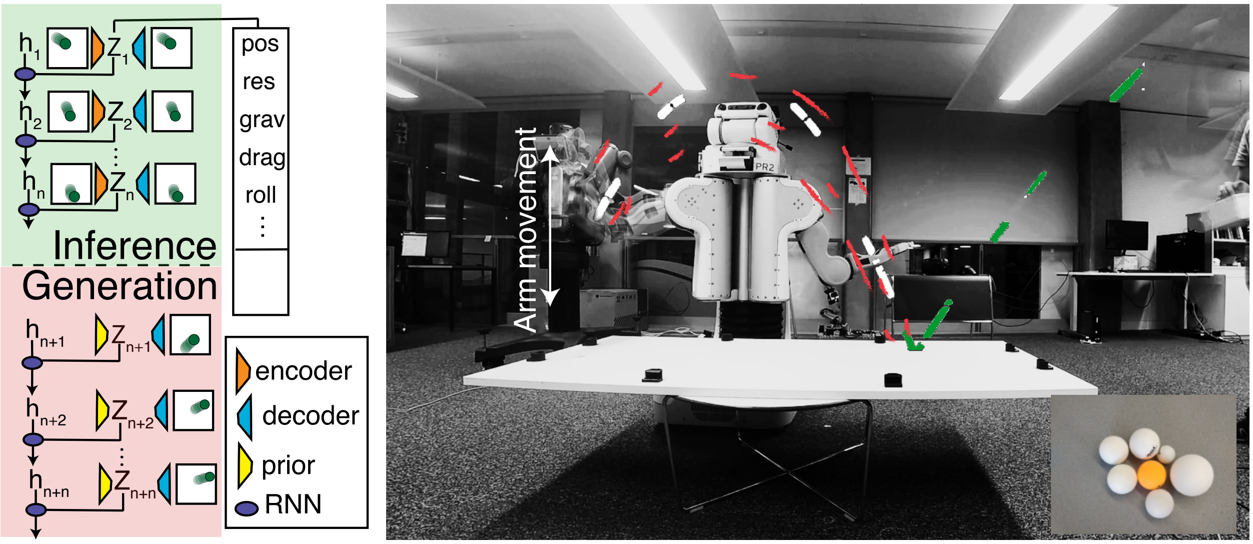


Figure 2: **Overview of and experimental setup.** We demonstrate the efficacy of our model by conducting experiments with a PR2 robot using its arm to intercept a bouncing ball mid-air, with different types of balls, each starting from a variety of different initial positions.

We are interested in making future predictions of the state evolution over time of a dynamic object, by conditioning the predictions on physical parameters. Given a set of sensory observations $x_{1:T}$, we are interested in predicting future observations $\overline{x}_{1:K}$ by using a low-dim representation $\overline{z}_{1:K}$ and the parameters of a governing physics model θ.

$$p\left(\overline{x}_{1:K}|x_{1:T}\right) = \int f_{dec}\left(\overline{x}_{1:K}|\overline{s}_{1:K}\right) p_{gen}\left(\overline{s}_{1:K}|s_{1:T},\theta\right) p_{inf}(\theta|s_{1:T}) f_{enc}\left(s_{1:T}|x_{1:T}\right) dz \tag{6}$$

We propose combining the encoder-decoder factorization with dynamics modelling in the latent space, conditioned on θ. We introduce an additional loss to the standard VRNN to encourage encoding of physically meaningful parameters, including a Gaussian negative log likelihood [29] loss between part of the latent space and the physical parameters θ we are interested in (e.g., gravity, restitution, position, and so on, in the case of a bouncing ball).

$$\begin{aligned}\mathbb{E}_{q(\mathbf{z}_{\le T}|\mathbf{x}_{\le T})}\Bigg[\sum_{t=1}^{T}\Bigg(& -\beta\mathrm{KL}\left(q\left(\mathbf{z}_t|\mathbf{x}_{\le t},\mathbf{z}_{<t}\right)\|p\left(\mathbf{z}_t|\mathbf{x}_{<t},\mathbf{z}_{<t}\right)\right) + \alpha\log p\left(\mathbf{x}_t|\mathbf{z}_{\le t},\mathbf{x}_{<t}\right) \\ & + \gamma\sum_{i}^{|\theta|}\left(\frac{1}{2}\ln(2\pi) + \frac{1}{2}\ln\left((\sigma^{i}_{enc,t})^2\right) + \frac{(\theta^i_t - \mu^i_{enc,t})^2}{2(\sigma^i_{enc,t})^2}\right)\Bigg)^2\Bigg]\end{aligned} \tag{7}$$

We can now perform probabilistic inference of physical parameters from sensory data such as a sequence of images,

$$P(\theta|x_{\le t}) = q\left(\mathbf{z}_t^{<|\theta|}|\mathbf{x}_{\le t},\mathbf{z}_{<t}\right) = \mathcal{N}(\boldsymbol{\mu}^{<[\theta|}_{enc,t},\boldsymbol{\sigma}^{<[\theta|}_{enc,t}) \tag{8}$$

by sampling future θ_{t+n} (eg. positions) and recursively updating the model predictions to generate possible future sensory states $P(\overline{x}_{\ge t}|x_{\le t},\theta)$.

Finally, we also modify the recurrent step by excluding x_t, since all the information is already present in z_t. This speeds up the prediction in the latent space, as x does not need to be reconstructed and fed back at every step.

$$\mathbf{h}_t = f_\theta\left(\varphi_\tau^{\mathbf{z}}\left(\mathbf{z}_t\right),\mathbf{h}_{t-1}\right) \tag{9}$$

To summarise, the contributions of this paper include:

1. Extension of the VRNN model with a loss to term to encode dynamical properties.
2. Enabling faster future predictions in the latent space, along with uncertainty quantification through the identified parameters.
3. Evaluation of speed and accuracy of identification, against alternate approaches to system identification
4. Demonstration on a physical robotic system, in a task requiring interception of a bouncing ball whose specific physical parameters are unknown a priori, requiring online identification from the video stream.

5 Experiments

First, we perform a series of experiments on simulated videos, when ground truth is explicitly available. We compare our method against existing system identification methods, evaluating speed of estimation and accuracy of the identified parameters. We then proceed to a physical experiment involving online system identification from a camera feed.

5.1 Setup

We use a bouncing ball as an example dynamical system. This is a particularly useful example, as the dynamics of the ball vary depending on the ball state, making system identification particularly challenging from high dimensional sensor data using classical techniques. The governing dynamics of the bouncing ball can be written down in terms of the following set of ordinary differential equations.

$$S = \begin{cases} x_t = x_0 + \dot{x}_0 t + \frac{1}{2}\ddot{x}t^2, \ddot{x} = g - d & \text{Free fall with air drag} \\ \dot{x}_t = -e\dot{x}_{t-1}, & \text{Bounce} \\ \dot{x}_t = r\dot{x}_{t-1}, & \text{Rolling} \end{cases} \tag{10}$$

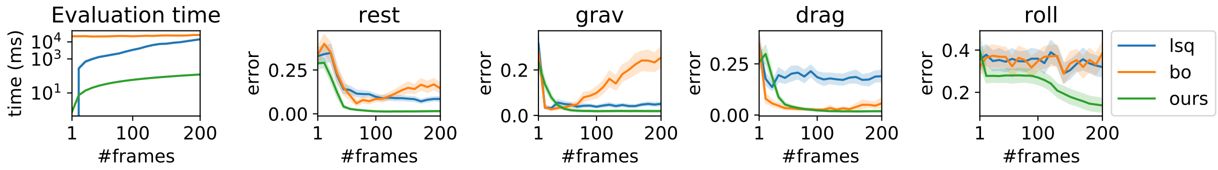
where $x, \dot{x}, \ddot{x}$ are the current position, velocity and acceleration respectively, e is the coefficient of restitution and r the rolling resistance. Additional dynamic effects are often observed such as air drag $d = -c\dot{x}s/m$ (s could be squared), where c is the drag constant, s is the current speed and m is mass. Thus the acceleration becomes $\ddot{x} = g + d$, where $g = -9.81$ is the gravitational force (but, in principle, could have a different magnitude). Thus the system is completely determined by the initial state of the system $x, \dot{x}, \ddot{x}$ and its physical properties $e, r, m, c, g \in \theta$. Of course the real world behaviour of any specific call could deviate from this model depending on its shape, initial spin (hence, Magnus effect), the presence of wind, and so on.

We use a parallel adaptive ODE solver to simulate data described by eq.10. We use these simulated trajectories to generate a sequence of images. We generate 10000 training and 100 test videos, with 200 timesteps/10 seconds, 28×28, with $e \in [0.6, 1.0]$, $g \in [-6.81, -12.81], d \in [0.05, 0.0005], r \in [0.0, 0.7]$. The baselines have access to the initial velocity, n number of positions and an optimized ODE solver for sampling. Our trained model receives *only* the video as an input, and no other parameters.

For the robot experiment, we trained a separate model with 5000 videos, 100×50 with 75 timesteps/10 seconds and the same physical parameters. Additionally, we add motion blur based on the velocity and black-out part of the frames to account for some of the missing/noisy data typically exhibited when using low-cost cameras. Additionally, we randomize the height of the plane on which the ball bounces. Our encoder-decoder network follow similar architecture as [12] and for the RNN we use a standard LSTM network. We set $\alpha = 1$, $\beta = 1$ and $\gamma = 10$ throughout our experiments (eq.7). We use an NVidia 1080 Ti for training and laptop NVidia Quadro M2000M GPUs for testing the model.

5.2 System identification

In this experiment, we evaluate the speed and accuracy of the proposed method against least-squares fitting [30] and using Bayesian Optimization [18] with a Gaussian Process prior over the parameters and an Expected Improvement acquisition function for sampling. Speed and accuracy benchmarks are shown in fig.3. It is clear that the proposed approach is both faster and more effective than these baselines.



(a) Speed

(b) Accuracy

Figure 3: **System Identification.** Performance of different system identification methods with variable number of observed frames. (a) Overall error of the predicted parameters (b) Speed of computation.

5.3 Forward predictions

Here, we evaluate the future prediction accuracy as frames are observed. Three sets of predictions are evaluated - after 20, 40 and 60 frames respectively - until the end of the video at 200 frames, as shown in fig.5. We visualize example model predictions and their associated uncertainty in fig. 4. In addition to previous baselines, the proposed model also outperforms a non-parametric model for system identification [26]. Importantly, the proposed approach becomes more certain as additional frames are observed, highlighting the probabilistic nature of Vid2Param.

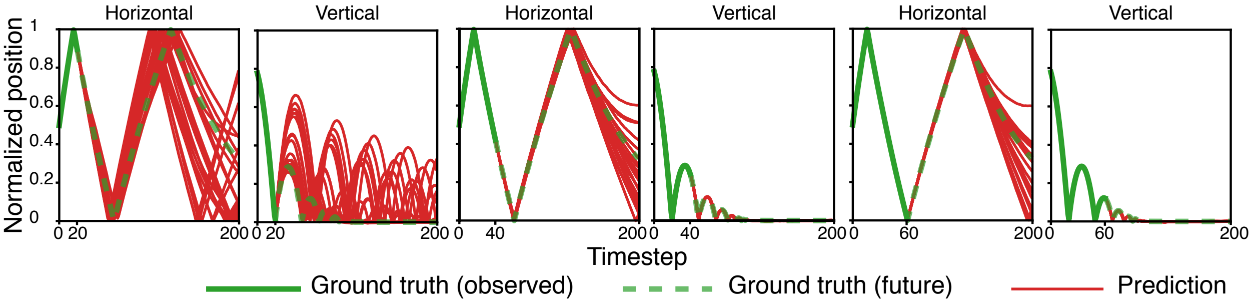
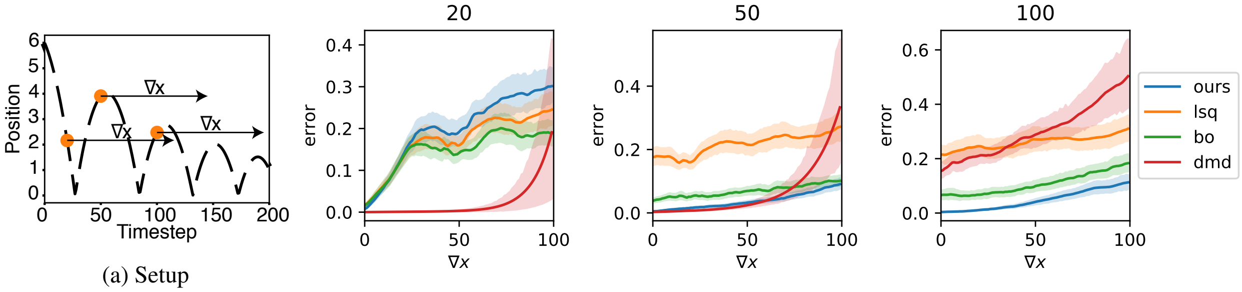


Figure 4: **Forward predictions uncertainty.** Associated uncertainty after different number of observed frames (20, 40 and 60). Observed trajectory (green), future ground truth (dashed-green), predictions (red).



(a) Setup

(b) Accuracy

Figure 5: **Accuracy of forward prediction.** Accuracy of the forward predictions in comparison to baselines. (a) Setup of the experiment - the accuracy is evaluated after 20, 50 and 100 frames are observed, predicting for the next 100 frames (b) Accuracy of the forward predictions. The DMD error is scaled down 10k, 500 and 5 times respectively for predictions after 20, 50 and 100 observations.

5.4 Varying physical properties of the object

In this experiment, we evaluate how well Vid2Param can estimate physical parameters when they are changing as the video is unrolled (we still use the same trained model on which the parameters stay constant throughout the video). Therefore, this is a test of robustness or sensitivity of the model. We generate a new dataset, wherein the parameters change some number of times within a single video - every 50 frames/2.5 seconds. The results are shown in fig.6. The results show that the proposed model can infer changing parameters, provided there is enough system excitation to facilitate this. For example, gravitation coefficients can only be inferred if the ball is bouncing.

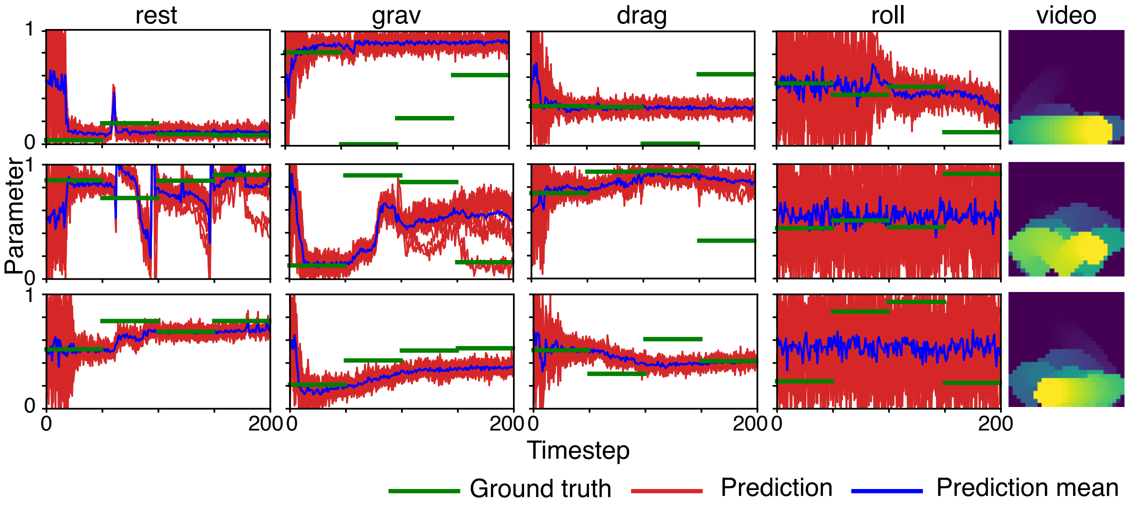
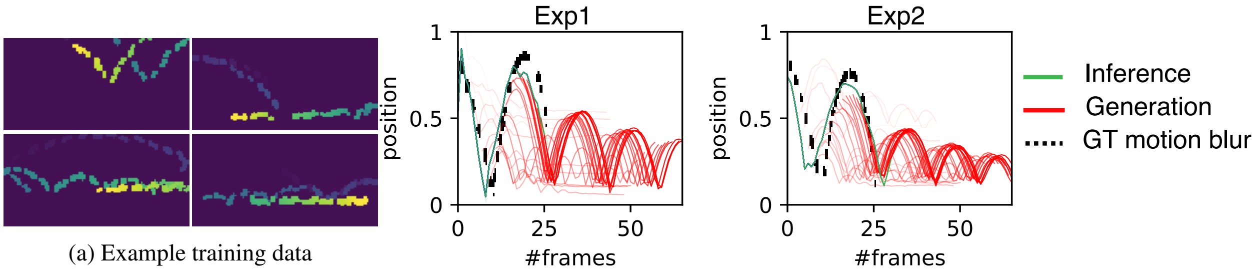


Figure 6: **System Identification from video with varying physical parameters.** The physical parameters of the bouncing ball change every 50 frames. We plot ground truth (green), predicted samples (red), and the predicted - mean (blue)

5.5 PR2 Robot experiments

Finally, we evaluate the accuracy and speed of our method in an experiment where the PR2 robot uses its arm to intercept a bouncing ball from a visual feed, using a standard low-cost camera as sensory input (please refer to supplementary video[1]). Firstly, the camera is calibrated with respect to the arm movements, so that predictions of the ball in the image, correspond to the same position of the gripper. No calibration with respect to the bouncing surface, position/velocity mappings, size of the ball, etc. are needed since these should be robustly dealt with by the model trained on randomized physics in simulation. The difference between two consecutive frames are fed directly into our model and the latent predictions are unrolled until the future predicted horizontal position is approximately the same as the horizontal position of the gripper of the arm. Then the generated vertical position of the ball is sent as a positional set point to the arm. An experimental run usually lasts for 2-3 seconds, during which the PR2 robot must infer the physics of the ball, predict its future trajectory and execute an action to intercept it. Our model runs at 20Hz on a standard laptop GPU, using IKFast for inverse kinematics of the arm. Examples of training data samples and future predictions can be seen in fig.7.

[1] https://youtu.be/8ZdlFQ8FM1A



(a) Example training data

(b) Example runs on PR2

Figure 7: **Robot experiments.** (a) Example of overlaid motion blur from the training videos. The height of the surface for the bounce, size, physics parameters, initial condition, strength of motion blur, etc. are randomized (b) Example observed data and model predictions. Trajectories are generated after every observation - the higher the intensity of the color, the more recent the prediction. None of the exact physical parameters of the ball are known *a priori*, e.g., the size of the ball, or the height of the table, but the model (trained only in simulation) is able to infer expected ball motion. (Please refer to supplementary video for additional results.)

6 Results and Analysis

System identification. We observe that the proposed method can accurately infer different physical parameters, outperforming baselines from the literature. The strength of gravity and air drag can usually be inferred from observing just a few frames. Air drag can usually be inferred after observing a few more frames, as it is a function of both horizontal and vertical velocity, rather than just the vertical velocity as in the case of gravity. Restitution factor can be inferred a few frames after the ball has bounced for the first time. The rolling coefficient has a high error, which starts to decrease towards the end of the videos. Our method can accurately estimate parameters with similar effect on the dynamics (gravity and air drag), as well as parameters whose effects are not observed until the end of the trajectory (rolling coefficient), where competing methods seem to struggle. Moreover, we also demonstrate that to an extent we can detect change in the parameters, as a video is unrolled, although this requires system excitation. This speaks to the robustness of this approach in practical field deployment.

Forward predictions. We have shown that our model can perform forward predictions in the latent space, over parameters of interest such as physical state variables. The forward predictions bring out key aspects of the evolution of uncertainty, such as high variability before a bounce and lower variability soon after, high variability over the stopping point before rolling is observed, etc. The proposed approach outperforms both parametric and non-parametric baselines in its ability to accurately perform forward predictions.

Limitations and future work. We observe in our robotics experiments that our model performs well in real settings. Nevertheless, we experienced some limitations arising from making the predictions based on a single image, e.g. the ball passing behind or in front of the gripper. Thus in the future in will be beneficial to extend this line of work by inferring future predictions from multiple sources of video stream from different locations.

Importantly, the proposed approach was able to generalise well to images captured from a real camera, despite only being trained on simulated data. This highlights the potential value of sim2real techniques for interpreting physical parameters in various applications, and its potential to enable reasoning about physical properties from relatively low fidelity sensor information that is bootstrapped by being grounded in learning from simulation.

7 Conclusions

In this paper, we present a method for online system identification from video. We benchmark our approach against baselines from the literature, outperforming them both in terms of speed and accuracy of identification. We then demonstrate the utility of this approach with the task of stopping a bouncing ball with a robot arm, performing online identification from a camera feed and using the proposed model for inference of the physics parameters. Further, we show the ability to generate future predictions of the ball position, laying the groundwork for much more sophisticated predictive motion planning schemes.

References

[1] A. Achille and S. Soatto. A separation principle for control in the age of deep learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:287–307, 2018.

[2] A. Ajay, J. Wu, N. Fazeli, M. Bauza, L. P. Kaelbling, J. B. Tenenbaum, and A. Rodriguez. Augmenting physical simulators with stochastic neural networks: Case study of planar pushing and bouncing. In *IROS*, pages 3066–3073. IEEE, 2018.

[3] D. Angelov, Y. Hristov, and S. Ramamoorthy. Using causal analysis to learn specifications from task demonstrations. AAMAS '19, pages 1341–1349, 2019.

[4] I. Arnekvist, D. Kragic, and J. A. Stork. Vpe: Variational policy embedding for transfer reinforcement learning. *ICRA*, 2019.

[5] P. Battaglia, R. Pascanu, M. Lai, D. J. Rezende, et al. Interaction networks for learning about objects, relations and physics. In *NeurIPS*, pages 4502–4510, 2016.

[6] R. Brincker, L. Zhang, and P. Andersen. Modal identification from ambient responses using frequency domain decomposition. In *International Modal Analysis Conference (IMAC)*, 2000.

[7] R. Brincker, L. Zhang, and P. Andersen. Modal identification of output-only systems using frequency domain decomposition. *Smart materials and structures*, 10(3):441, 2001.

[8] S. L. Brunton, J. L. Proctor, and J. N. 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.

[9] M. Chang, T. Ullman, A. Torralba, and J. B. Tenenbaum. A compositional object-based approach to learning physical dynamics. In *ICLR*, 2017.

[10] Y. Chebotar, A. Handa, V. Makoviychuk, M. Macklin, J. Issac, N. Ratliff, and D. Fox. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. *ICRA*, 2019.

[11] S. Chen, S. Billings, and P. Grant. Non-linear system identification using neural networks. *International journal of control*, 51(6):1191–1214, 1990.

[12] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NeurIPS*, pages 2172–2180, 2016.

[13] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio. A recurrent latent variable model for sequential data. In *NeurIPS*, pages 2980–2988, 2015.

[14] F. de Avila Belbute-Peres, K. Smith, K. Allen, J. Tenenbaum, and J. Z. Kolter. End-to-end differentiable physics for learning and control. In *NeurIPS*, pages 7178–7189, 2018.

[15] J. Degrave, M. Hermans, J. Dambre, et al. A differentiable physics engine for deep learning in robotics. *Frontiers in neurorobotics*, 13, 2019.

[16] S. A. Eslami, N. Heess, T. Weber, Y. Tassa, D. Szepesvari, G. E. Hinton, et al. Attend, infer, repeat: Fast scene understanding with generative models. In *NeurIPS*, pages 3225–3233, 2016.

[17] M. Fraccaro, S. Kamronn, U. Paquet, and O. Winther. A disentangled recognition and nonlinear dynamics model for unsupervised learning. In *NeurIPS*, pages 3601–3610, 2017.

[18] J. González. Gpyopt: a bayesian optimization framework in python, 2016.

[19] Y. Hristov, A. Lascarides, and S. Ramamoorthy. Interpretable latent spaces for learning from demonstration. In *CoRL*, volume 87, pages 957–968. PMLR, 29–31 Oct 2018.

[20] J.-T. Hsieh, B. Liu, D.-A. Huang, L. F. Fei-Fei, and J. C. Niebles. Learning to decompose and disentangle representations for video prediction. In *NeurIPS*, pages 515–524, 2018.

[21] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *NeurIPS*, pages 2017–2025, 2015.

[22] M. Jaques, M. Burke, and T. Hospedales. Physics-as-inverse-graphics: Joint unsupervised learning of objects and physics from video. *arXiv preprint arXiv:1905.11169*, 2019.

[23] D. Jayaraman, F. Ebert, A. A. Efros, and S. Levine. Time-agnostic prediction: Predicting predictable video frames. *arXiv preprint arXiv:1808.07784*, 2018.

[24] J.-N. Juang and R. S. Pappa. An eigensystem realization algorithm for modal parameter identification and model reduction. *J GUID CONTROL DYNAM*, 8(5):620–627, 1985.

[25] M. Karl, M. Soelch, J. Bayer, and P. van der Smagt. Deep variational bayes filters: Unsupervised learning of state space models from raw data. *arXiv preprint arXiv:1605.06432*, 2016.

[26] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor. *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM, 2016.

[27] Z. Ma, S. Ahuja, and C. W. Rowley. Reduced-order models for control of fluids using the eigensystem realization algorithm. *THEOR COMP FLUID DYN*, 25(1-4):233–247, 2011.

[28] D. Mrowca, C. Zhuang, E. Wang, N. Haber, L. F. Fei-Fei, J. Tenenbaum, and D. L. Yamins. Flexible neural representation for physics prediction. In *NeurIPS*, pages 8813–8824, 2018.

[29] D. A. Nix and A. S. Weigend. Estimating the mean and variance of the target probability distribution. In *ICNN'94*, volume 1, pages 55–60. IEEE, 1994.

[30] G. Pavlak. Comparison of bayesian parameter estimation and least squares minimization for inverse grey-box building model identification.". 2011.

[31] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *ICRA*, pages 1–8. IEEE, 2018.

[32] V. Peterka. Bayesian approach to system identification. In *Trends and Progress in System identification*, pages 239–304. Elsevier, 1981.

[33] S. Purushwalkam, A. Gupta, D. M. Kaufman, and B. Russell. Bounce and learn: Modeling scene dynamics with real-world bounces. *ICLR*, 2019.

[34] F. Ramos, R. C. Possas, and D. Fox. Bayessim: adaptive domain randomization via probabilistic inference for robotics simulators. *arXiv preprint arXiv:1906.01728*, 2019.

[35] D. Romeres, G. Prando, G. Pillonetto, and A. Chiuso. On-line bayesian system identification. In *2016 European Control Conference (ECC)*, pages 1359–1364. IEEE, 2016.

[36] C. W. Rowley. Model reduction for fluids, using balanced proper orthogonal decomposition. *International Journal of Bifurcation and Chaos*, 15(03):997–1013, 2005.

[37] M. G. Safonov and R. Chiang. A schur method for balanced-truncation model reduction. *IEEE Transactions on Automatic Control*, 34(7):729–733, 1989.

[38] C. Schenck and D. Fox. Spnets: Differentiable fluid dynamics for deep neural networks. In *CoRL*, pages 317–335, 2018.

[39] G. Shi, X. Shi, M. O'Connell, R. Yu, K. Azizzadenesheli, A. Anandkumar, Y. Yue, and S.-J. Chung. Neural lander: Stable drone landing control using learned dynamics. *ICRA*, 2019.

[40] N. Watters, D. Zoran, T. Weber, P. Battaglia, R. Pascanu, and A. Tacchetti. Visual interaction networks: Learning a physics simulator from video. In *NeurIPS*, pages 4539–4547, 2017.

[41] J. Wu, I. Yildirim, J. J. Lim, B. Freeman, and J. Tenenbaum. Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. In *NeurIPS*, pages 127–135, 2015.

[42] A. Zeng, S. Song, J. Lee, A. Rodriguez, and T. Funkhouser. Tossingbot: Learning to throw arbitrary objects with residual physics. *arXiv preprint arXiv:1903.11239*, 2019.

[43] T. Zhang, G. Kahn, S. Levine, and P. Abbeel. Learning deep control policies for autonomous aerial vehicles with mpc-guided policy search. In *ICRA*, pages 528–535. IEEE, 2016.