Analysis of ODE2VAE with Examples

Batuhan Koyuncu*
Department of Computer Science
Saarland University
Saarbrücken, Germany
koyuncu@cs.uni-saarland.de

Abstract

Ordinary Differential Equation Variational Auto-Encoder (ODE2VAE) is a deep latent variable model that aims to learn complex distributions over high-dimensional sequential data and their low-dimensional representations in a hierarchical latent space. The hierarchical organization of the latent space embeds a physics-guided inductive bias in the model. In this paper, we analyze the latent representations inferred by the ODE2VAE model over three different physical motion datasets: bouncing balls, projectile motion, and simple pendulum. We show that the model is able to learn meaningful latent representations to an extent without any supervision.

1 Introduction

ODE2VAE is a dynamic generative model that operates in continuous-time [1]. It aims to learn complex latent trajectories of sequential data by combining VAEs, Neural ODEs, and Bayesian neural networks (BNNs) [2]. The ODE2VAE model infers the latent trajectories by using coupled latent ODEs which induce a physics-motivated inductive bias into the model. It was shown that the ODE2VAE model exceeds the performance of the compared models in the task of extrapolating future time steps of sequential datasets [1]. However, the latent representations learned by the model are not explored. Given its physics-motivated inductive bias, the ODE2VAE model may learn meaningful latent representations behaving according to approximate physical generating factors of a physical system. Our first contribution is investigating ODE2VAE's performance on modeling the three motion datasets: bouncing balls, projectile motion, and simple pendulum. Moreover, we uncover the effects of the model's by analyzing the model's dynamical latent representations and their uncertainties.

2 Methods

ODE2VAE proposes a second order latent ODE model to infer latent representations of high-dimensional sequential data such as videos [1]. The ODE2VAE model uses VAEs for encoding the input frames and coupled first order latent ODEs for modeling the latent trajectories. This approach creates an hierarchical latent space. Therefore, the model's latent space can be decomposed into latent position $\mathbf{s}_t \in \mathbb{R}^a$, velocity $\mathbf{v}_t \in \mathbb{R}^a$, and acceleration trajectories $\mathbf{f}_{\mathcal{W}} \in \mathbb{R}^a$ as the following:

$$\begin{bmatrix} \mathbf{s}_0 \\ \mathbf{v}_0 \end{bmatrix} \sim q_{\text{enc}}\left(\begin{pmatrix} \mathbf{s}_0 \\ \mathbf{v}_0 \end{pmatrix} \mid \mathbf{x}_{0:N}\right) \tag{1}$$

$$\underbrace{\begin{bmatrix} \mathbf{s}_t \\ \mathbf{v}_t \end{bmatrix}}_{\mathbf{z}_t} = \begin{bmatrix} \mathbf{s}_0 \\ \mathbf{v}_0 \end{bmatrix} + \int_0^t \begin{bmatrix} \mathbf{v}_\tau \\ \mathbf{f}_{\mathcal{W}}(\mathbf{s}_\tau, \mathbf{v}_\tau) \end{bmatrix} d\tau, \quad \dot{\mathbf{s}}_t = \mathbf{v}_t \qquad \dot{\mathbf{v}}_t = \mathbf{f}_{\mathcal{W}}(\mathbf{s}_t, \mathbf{v}_t) \tag{2}$$

*Work done while at Boğaziçi University, İstanbul, Turkey.

Fourth Workshop on Machine Learning and the Physical Sciences (NeurIPS 2021).

where $\mathbf{x}_{0:N}$ denotes the sequential input data, q_{enc} denotes approximate posterior distribution for the initial latent state and is parameterized by CNNs, and the acceleration field $\mathbf{f}_{\mathcal{W}}$ is parameterized by a BNN with weights $\mathcal{W}$. The latent state $\mathbf{z}_t$ corresponds to the tuple $[\mathbf{s}_t, \mathbf{v}_t]$. The hierarchy among the trajectories imposes the physics-guided inductive bias of the model, since the trajectories model an arbitrary equation of motion in the latent space by using latent ODEs. In other words, the latent velocity trajectory drives the dynamics of the position trajectory and the latent acceleration field drives the latent velocity trajectory. Therefore, one can expect the model to learn decomposed latent trajectories that are aligned with the real dynamics (i.e. observed position, velocity, acceleration) of the input sequence.

The ODE2VAE model is optimized by using variational inference. The ELBO term, which is the lower bound of the marginal log-likelihood, can be written as [1]:

$$\log p(X) \geq \underbrace{-\mathcal{D}_{KL}[q(\mathcal{W}, Z \mid X) || p(\mathcal{W}, Z)]}_{\text{Regularization}} + \underbrace{\mathbb{E}_{q(\mathcal{W},Z|X)}[\log p(X \mid \mathcal{W}, Z)]}_{\text{Reconstruction}} \tag{3}$$

$$+ \underbrace{\sum_{i=1}^{N} \mathbb{E}_{q_{\text{enc}}(\mathbf{z}_0|X)} \left[\mathbb{E}_{q_{\text{ode}}(\mathcal{W},\mathbf{z}_i|X,\mathbf{z}_0)} \left[-\log \frac{q_{\text{ode}}\left(\mathbf{z}_i \mid \mathbf{z}_0, \mathcal{W}\right)}{p\left(\mathbf{z}_i\right)} + \log p\left(\mathbf{x}_i \mid \mathbf{z}_i\right) \right] \right]}_{\text{dynamic loss}} \tag{4}$$

where X and Z denote the sequential data $\mathbf{x}_{0:N}$ and sequence of latent variables $\mathbf{z}_{0:N}$ respectively. The amortized approximate posterior distributions over the weights of the BNN $q(\mathcal{W})$ and initial latent state $q_{\text{enc}}(\mathbf{z}_0)$ follow a multivariate Gaussian distribution. The approximate posterior over the next latent states $q_{\text{ode}}(\mathbf{z}_i)$ is computed using continuous normalizing flow given the initial log density [3]. The priors $p(\mathcal{W})$, $p(\mathbf{z}_0)$, and $p(\mathbf{z}_{1:N})$ are standard multivariate Gaussians.

To investigate the effects of the imposed inductive bias over the decomposed latent space, we analyze L2 norms of the latent states as they are previously used for explanatory and interpretable metrics [4, 5]. Due to the model's inductive bias, the L2 norm of the acceleration field is similar to the magnitude of the force effecting the dynamics in the latent space. Similarly, the L2 norm of the velocity latent variable resembles square root of the total kinetic energy of the system.

3 Experiments

3.1 Data and Training

For each dynamical system, we follow the provided implementation for data generation [6]. Our data consists of sequences of $32 \times 32 \times 1$ images with pixel values between 0 and 1. For each dataset, the number of cases are 10000, 500, 500 for training, validation, and test sets. Other details about the datasets can be found in Appendix A.1. All of the model variants are created by using the official ODE2VAE implementation and hyperparameters for bouncing balls experiments [1] in Tensorflow framework [7]. The models are trained with Adam Optimizer [8] with the learning rate of $\eta = 0.001$ and the batch size of 32. The amortized inference length is selected as $m = 3$, which is the input length. All hyperparameters are chosen according to the baseline work's setting for bouncing ball experiments [1]. Our experiments are executed on a single Tesla V100 GPU where each experiment takes approximately three days. We start the experiments with the minimum possible number of latent units for the given dataset and increase the dimensionality until the motion is captured. This approach creates a bottleneck in the latent dimensionality. It forces the model to learn meaningful latent representations because the ELBO objective can only be maximized by the latent units that are related with the ground truth generative factors [9].

3.2 Results

The experiment results are evaluated by computing L2 norms of the latent states, mean squared error (MSE), peak signal-to-noise ratio (PSNR), and negative marginal log-likelihood (NLL). In the experiments, we use L=10, which is the number of latent states sampled at each time step.

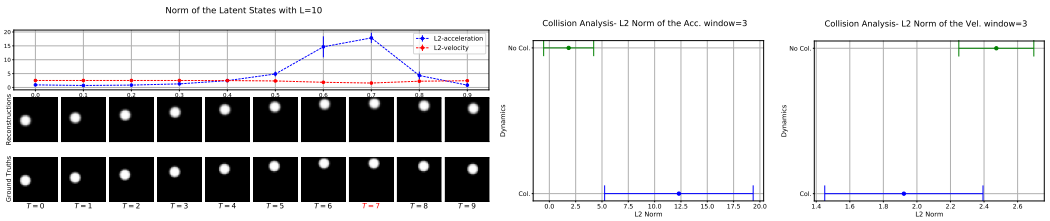
3.2.1 Bouncing Balls

Bouncing ball motion consists of a moving ball with a constant kinetic energy in a 2D box. When the ball hits the wall, the wall exerts a force over the ball proportional to the change in the ball's momentum. Therefore, the ball has a non-zero acceleration only for the collision moments. In this experiment, we check if the ODE2VAE model captures latent representations that are aligned with real dynamics. The baseline model with the latent dimensionality $a = 2$ cannot capture the dynamics of the motion of the single bouncing ball after it is trained for 250 epochs. We increase the latent dimensionality to $a = 3$ and $a = 12$ and train the models for 250 epochs. Both of the models have captured the dynamics of the single bouncing ball.

Table 1: Performance metrics of the selected model on the single bouncing ball. Each metric is computed by using 10 samples per test case.

Model	Metrics		
	MSE	PSNR	NLL
ODE2VAE, $a = 3$	0.0027 ± 0.0032	28.2601 ± 4.6264	42.5142

In Table 1, we only report the metrics for model with $a = 3$. The model is able to capture physically meaningful latent representations. We present an example case in Figure 1(a). When the ball hits the wall, there is a spike in the norm of the latent acceleration. It can be seen that the standard deviation of the norm of the acceleration field also increases during the collision. This indicates the fact that the output of the BNN has a greater uncertainty during the collision. The norm of the latent velocity is not changed during the motion in the example case, which suggests the fact that the latent velocity preserves its norm, but changes the direction during the collisions (see Figure 5). Figures 1(b) and 1(c) summarize the statistics for the norm of the acceleration field and latent velocity over all test cases with a breakdown for the time steps with and without collision. Figure 1(b) shows that the model generates an acceleration field with a greater magnitude during the collisions. When there no collision, the model generates an acceleration field with a smaller norm compared to the mean magnitude at the collision time points. Figure 1(c) shows that the model has increased the magnitude of the velocity latent without collision. Its magnitude decreases during the collision moments, which is unexpected since the ball's stationary moments during the collisions are ignored in the dataset.



(a) (b) (c)

Figure 1: (a) Example test case reconstructed by the ODE2VAE model with $a = 3$ and L2 norms of the latent states. Mean and standard deviation values for the L2-norm of the latent (b) acceleration and (c) velocity of the bouncing ball dataset.

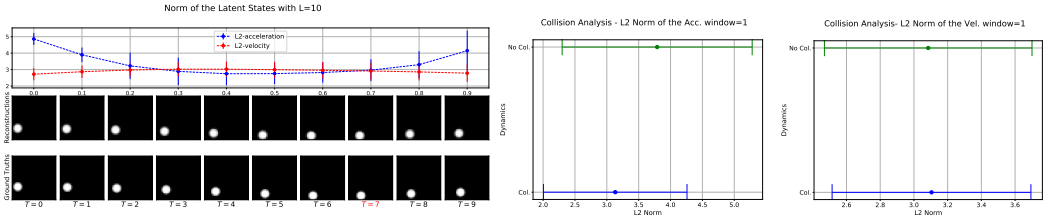
3.2.2 Projectile Motion

Projectile motion describes an object's motion due its initial velocity and the gravitational force. Our sequences captures a ball's motion which is projected into the air. The ball is under a constant gravitational acceleration during its motion. It is expected to accelerate as it approaches to the floor. As it hits the ground, it loses a fraction of its total energy and bounces back. Therefore, the ball is under constant acceleration except the collision times. Also, it has the minimum kinetic energy when it is at the maximum height. We check if the ODE2VAE model learns physically meaningful representation due to its hierarchical latent space. We find out that the ODE2VAE baseline model could not capture the projectile motion dynamics with $a = 2, 3, 5, 7$ after it is trained for 300 epochs and converged. Therefore, we increased the number of latent units to 9. Although the model has increased predictive performance with $a = 9$, it is not fully able to capture latent dynamics that resemble the real projectile motion.

We present an example case in Figure 2(a). In this dataset, collisions between the ball and floor is assumed to take 0.1 second. The frames are separated by 0.1 second. One of the challenges about the projectile motion is that the model needs to capture a constant gravitational acceleration present at all times as well as the instantaneous acceleration due to the collision. The results show that the model is not able to generate a constant acceleration field for the moments without collision. On the other hand, there is a slight increment in the total kinetic energy when the ball is closer to the ground. In Figures 2(b) and 2(c), we present the norm of the acceleration and velocity latents at the time steps with and without collision. Although the true acceleration field is constant (except for the collision moments) due to the constant free-fall acceleration, the model is not able to learn a fixed acceleration field among the different test cases (see Figure 2(b)). We omit to comment on the norm of the latent velocity since the ground truth velocities are not the same in the test set. We should note that another reason behind model's inability to capture projectile motion's dynamics may be time resolution of the input sequences. Since the time step between frames have the same length with the time spent during collisions, the model may tend to overfit at the collision moments. This behaviour can be analyzed by the use of an input data with higher temporal resolution.

Table 2: Performance metrics of the selected model on the projectile motion dataset. Each metric is computed by using 10 samples per test case.

	Metrics		
Model	MSE	PSNR	NLL
ODE2VAE, $a = 9$	0.0016 ± 0.0016	30.5322 ± 5.1913	27.8267



(a) (b) (c)

Figure 2: (a) Example test case reconstructed by the ODE2VAE model with $a = 9$ and L2 norms of the latent states. Mean and standard deviation values for the L2-norm of the latent (b) acceleration and (c) velocity of the projectile motion dataset.

3.2.3 Simple Pendulum

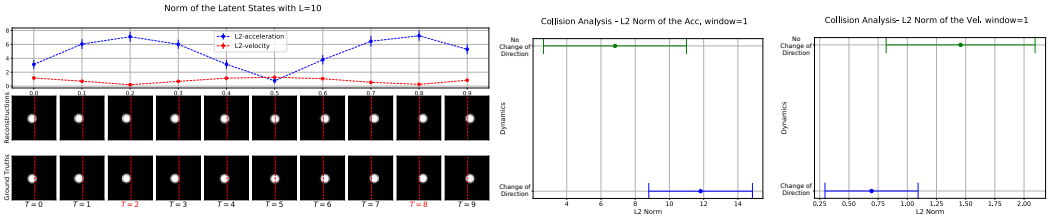
Simple pendulum motion consists of an moving mass which is suspended from a rod that is pivoted. It can simulated by small angle approximation and the motion becomes periodic and it can simulated by small angle approximation, when the air friction is neglected. During the motion, the object is under the effect of uniform gravitational field. It has zero kinetic energy when its at the highest point of swing. It reaches maximum kinetic energy at its equilibrium position. The object also effected by the force applied by the rod; the magnitude of the restoring force over the object increases as it approaches its highest point of swing. Since the motion can be described with a second-order differential equation, we can check if the ODE2VAE model learn meaningful latent representations. We train the ODE2VAE model with $a = 2$ and $a = 6$ on the simple pendulum dataset for 300 epochs. Both models have captured the pendulum motion; there is not a considerable improvement between the two models.

In Table 3, we only report the model's performance with $a = 2$. Since the simple pendulum motion is a periodic motion, it may be easier to capture compared to the other motion types. The model is able to capture physically meaningful latent representations. We provide an example case in Figure 3(a). The model generates an increased norm of the latent acceleration when the object reaches its highest

Table 3: Performance metrics of the selected model on the simple pendulum dataset. Each metric is computed by using 10 samples per test case.

	Metrics		
Model	MSE	PSNR	NLL
ODE2VAE, $a = 2$	0.0007 ± 0.0006	33.6325 ± 4.5241	26.5609

point during its motion and the magnitude of the acceleration is minimized when the object passes through the equilibrium point. Another observation is that the BNN has the decreased uncertainty over the magnitude of the acceleration field when the object passes through the equilibrium point. Additionally, the norm of the latent velocity reaches its peak when the ball passes the equilibrium position and decreases when the object reaches its highest points, which are the highlighted time points. In Figure 3(b), we provide the statistics for the norm of the latent acceleration over the test cases with a breakdown for the moments with and without direction change. The figure shows that the model generates an acceleration field with a greater magnitude when the object reaches its highest point. Moreover, Figure 3(c) shows that the latent velocities of the test cases have a reduced norm at the moments of direction change. Although the model cannot generate latent velocity variables with magnitude zero at the moment of direction change, it captures a decreasing trend, which is still physically plausible.



(a) (b) (c)

Figure 3: (a) Example test case reconstructed by the ODE2VAE model with $a = 2$ and L2 norms of the latent states. Mean and standard deviation values for the L2-norm of the latent (b) acceleration and (c) velocity of the pendulum dataset.

4 Conclusion and Future Work

Our experiments showed that the ODE2VAE model is able to learn physically meaningful latent representations in an unsupervised setting. We observed that the model can learn physically plausible dynamic latent representations for the bouncing ball and simple pendulum datasets. Additionally, our results empirically show that the uncertainty over the magnitude of the acceleration field increased during rare events and non-linear motions such as collisions. Our work has limitations that stem from convergence issues during the model training and limited number of metrics for measuring the interpretability of the latent representations in an unsupervised setting. Therefore, our future work may focus on building a stable version of ODE2VAE and increasing the interpretability of the latent representations through using arbitarary Lagrangians or Hamiltonians [10, 11], and learning disentangled latent representations with weak supervision [12].

Broader Impact

Embedding the right inductive bias in the machine learning model is a key step for increasing generalization capacity of the network and its interpretability [13, 14]. This project will have broad impact in building interpretable latent representations and quantifying their robustness for dynamical systems with sequential data. As there are task specific inductive biases that increase model generalization and interpretability, we can also design metrics that will increase the interpretability of the latent representations and quantify the model's robustness. Although the data and metrics used in the experiments are from well-known physical motion datasets, they can be extended for other domains such as chemistry, biology, and drug discovery. Therefore, this approach may provide deep latent variable models which give insight to experts from various domains.

Acknowledgments

This paper is derived from Batuhan Koyuncu's MS thesis "Analysis and Regularization of Deep Generative Second Order Ordinary Differential Equations" completed in 2021 at Boğaziçi University under the supervision of Prof. Lale Akarun. This work was supported by Boğaziçi University Research Fund under the Grant Number 16903. We also thank Inzva for the computing resources provided.

References

[1] Cagatay Yildiz, Markus Heinonen, and Harri Lähdesmäki. ODE2VAE: deep generative second order odes with bayesian neural networks. In *Advances in Neural Information Processing Systems*, volume 32, pages 13412–13421. Curran Associates, Inc., 2019.

[2] Yarin Gal and Zoubin Ghahramani. Bayesian convolutional neural networks with bernoulli approximate variational inference. *CoRR*, abs/1506.02158, 2015.

[3] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 31, pages 6572–6583. Curran Associates, Inc., 2018.

[4] Yulia Rubanova, Tian Qi Chen, and David Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. In *Advances in Neural Information Processing Systems*, volume 32, pages 5321–5331. Curran Associates, Inc., 2019.

[5] Daehoon Gwak, Gyuhyeon Sim, Michael Poli, Stefano Massaroli, Jaegul Choo, and Edward Choi. Neural ordinary differential equations for intervention modeling. *CoRR*, abs/2010.08304, 2020.

[6] Ilya Sutskever, Geoffrey E. Hinton, and Graham W. Taylor. The recurrent temporal restricted boltzmann machine. In *Advances in Neural Information Processing Systems*, volume 21, pages 1601–1608. Curran Associates, Inc., 2008.

[7] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek Gordon Murray, Benoit Steiner, Paul A. Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zhang. Tensorflow: A system for large-scale machine learning. *CoRR*, abs/1605.08695, 2016.

[8] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations (ICLR)*, 2015.

[9] Marco Fraccaro, Simon Kamronn, Ulrich Paquet, and Ole Winther. A disentangled recognition and nonlinear dynamics model for unsupervised learning, 2017.

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

[11] Peter Toth, Danilo J. Rezende, Andrew Jaegle, Sébastien Racanière, Aleksandar Botev, and Irina Higgins. Hamiltonian generative networks. In *8th International Conference on Learning Representations (ICLR)*. OpenReview.net, 2020.

[12] Ricky T. Q. Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, volume 31, pages 2615–2625. Curran Associates, Inc., 2018.

[13] Anirudh Goyal and Yoshua Bengio. Inductive biases for deep learning of higher-level cognition. *CoRR*, abs/2011.15091, 2020.

[14] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019*, volume 97, pages 4114–4124. PMLR, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]

(c) Did you discuss any potential negative societal impacts of your work? [No] There is no potential negative societal impact.

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...

   (a) Did you state the full set of assumptions of all theoretical results? [N/A]

   (b) Did you include complete proofs of all theoretical results? [N/A]

3. If you ran experiments...

   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We followed the official implementation of ODE2VAE model[1]. The data and training instructions are reported in Sections 3.1 and A.1

   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 3.1.

   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] All error bars (uncertainties) are computed by using 10 samples per test case, see Section 3.2

   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 3.1

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

   (a) If your work uses existing assets, did you cite the creators? [Yes]

   (b) Did you mention the license of the assets? [N/A] Used assets are provided in open source without any license.

   (c) Did you include any new assets either in the supplemental material or as a URL? [No] We only describe them.

   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] We use synthetic data of physical systems.

   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] We use synthetic data of physical systems.

5. If you used crowdsourcing or conducted research with human subjects...

   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Appendix

A.1 Data Generation

A.1.1 Bouncing Balls

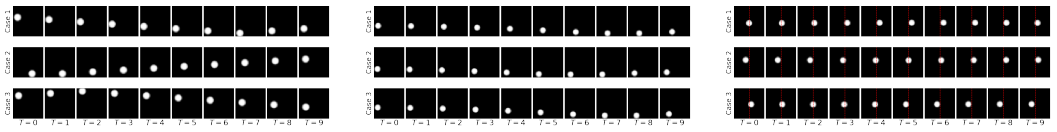
By using the provided implementation [6], we re-implemented the bouncing balls dataset with multiple variants. The dataset captures dynamics of a single ball in a 2D box. There is no friction in the motion and all collisions are elastic. The 2D box has a side length of $10.0\,\mathrm{m}$. The radius and mass of the ball are fixed, at $1.2\,\mathrm{m}$ and $1.0\,\mathrm{kg}$, respectively. The ball has a velocity that is randomly sampled from a standard normal distribution. The total kinetic energy is constant over each sequence. The frames in the sequences are separated by one second, and the ball's motion is simulated with 0.5 second resolution. In Figure 4(a), we present a figure of the bouncing ball dataset.

A.1.2 Projectile Motion

The projectile motion dataset consists of a ball, which is projected up in a square frame with the side length of $10.0\,\mathrm{m}$. The ball is affected by gravity and collisions during its motion. The ball reaches its maximum kinetic energy before its first collision with the ground. During the collision it loses some of its kinetic energy. It has a radius of $1.0\,\mathrm{m}$. The initial velocity $(\mathrm{m/s})$ is denoted as $v = [v_x, v_y]$, where v_x and v_y are sampled from a uniform distribution with the ranges $[1, 4]$ and $[0, 1]$. The initial position (m) is denoted as $h = [h_x, h_y]$, where h_x is fixed as zero and h_y has a uniform distribution in the range $[1, 3]$. The coefficient of restitution, which determines the magnitude of the v_y after the ball hits and bounces from the floor, is 0.80. The collision between the ball and floor is assumed to take 0.1 second. The frames are separated by 0.1 second. Figure 4(b) shows example sequences from the projectile motion dataset.

A.1.3 Simple Pendulum

A simple pendulum system consists of a point mass, and a pivoted rod with length l, where the mass is suspended from the rod. The mass of the rod is negligible, and there is no air friction in our case. In this paper, we only consider pendulum motions with small initial angles. Therefore, it is sufficient to model simple pendulum motion. The dynamics of the simple pendulum is approximated by using small angle approximation, $\sin\alpha \approx \alpha$. The dataset captures the periodic motion of the point mass around its equilibrium position. During the motion, the gravitational field is uniform. The object reaches its maximum kinetic energy at its equilibrium position. The magnitude of the restoring force over the object increases as it approaches its highest point of swing. The side length of the 2D square box is $10.0\,\mathrm{m}$. The radius of point mass is $1.0\,\mathrm{m}$. The length of the rod l has a uniform distribution in the range $[3, 6]$. The initial angle for freeing the point mass follows a uniform distribution in the range $[\pi/36, \pi/9]$. The gravitational field has a magnitude of $9.91\,\mathrm{m/s^2}$. The frames are separated by 0.4 seconds, and the motion is simulated by the analytical solution for the simple pendulum motion. Due to the different initial angles, the maximum total kinetic energy for each case is different. Figure 4(c) shows example sequences from the pendulum dataset.



(a) Bouncing ball dataset. (b) Projectile motion dataset. (c) Simple pendulum dataset.

Figure 4: Dataset figures.

A.2 Additional Results

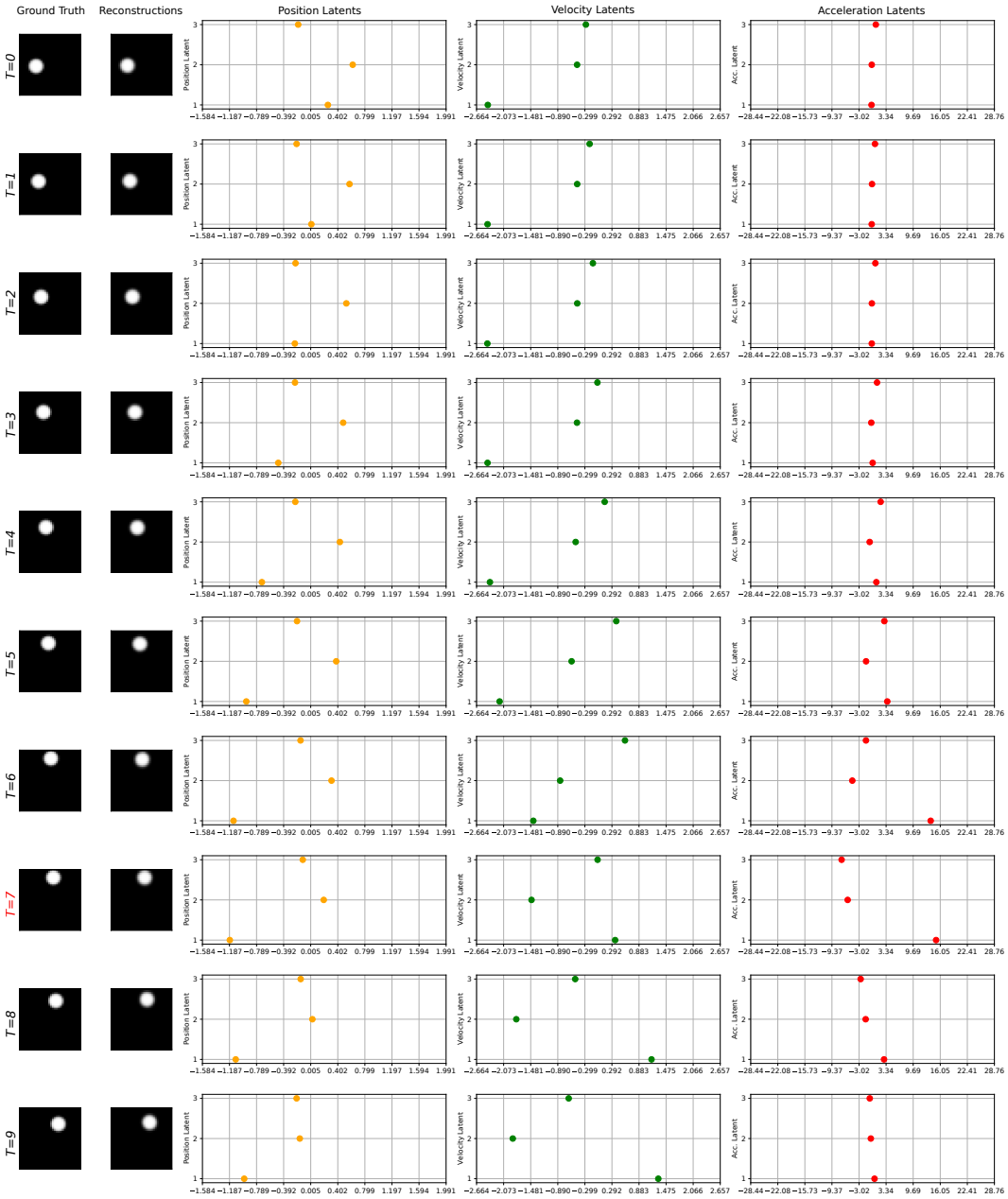


Figure 5: Example test case for single bouncing ball with corresponding latent vectors. The figure explicitly shows the evolution of the latent vectors.