Accepted for publication in German Conference on Pattern Recognition (GCPR) 2019, to appear

Learning to Disentangle Latent Physical Factors for Video Prediction

Deyao Zhu[1 2], Marco Munderloh[2], Bodo Rosenhahn[2], Jörg Stückler[1]

[1]Max Planck Institute for Intelligent Systems [2]Leibniz Universität Hannover

Abstract. Physical scene understanding is a fundamental human ability. Empowering artificial systems with such understanding is an important step towards flexible and adaptive behavior in the real world. As a step in this direction, we propose a novel approach to physical scene understanding in video. We train a deep neural network for video prediction which embeds the video sequence in a low-dimensional recurrent latent space representation. We optimize the total correlation of the latent dimensions within a variational recurrent auto-encoder framework. This encourages the representation to disentangle the latent physical factors of variation in the training data. To train and evaluate our approach, we use synthetic video sequences in three different physical scenarios with various degrees of difficulty. Our experiments demonstrate that our model can disentangle several appearance-related properties in the unsupervised case. If we add supervision signals for the latent code, our model can further improve the disentanglement of dynamics-related properties.

1 Introduction

A fundamental ability of humans for understanding dynamic scenes is to perceive physical properties of objects and predicting the physical evolution of a scene coarsely into the future. Providing cyber-physical systems with these abilities is a key ingredient to flexible and adaptive behavior in the real word. A large body of computer vision research has recently demonstrated the success of deep learning techniques for tasks such as object detection and recognition in images or video prediction. Learning to reason about the dynamic physical states of objects in video attracts increasing attention recently. A significant part of this research focuses on regressing the physical states of the system from images and using a physics-engine-like module to predict successive frames [28, 2, 25, 30]. Although this is a straightforward approach, it requires hand-crafted tailoring of the state representation and simulator for the specific task. For example, one needs to decide the physical laws to use or the number of represented objects. Some studies instead directly predict future frames end-to-end using deep learning based models [31]. Learning latent state representations that disentangle the physical factors of variation in the data such as object speed, position, mass, and friction, however, is still an open research problem. Such models would allow for introspection of the physical properties of a scene.

In this paper, we propose a variational approach to video prediction that learns a recurrent latent representation of the video and allows for predicting sequences into the future. Our network architecture is inspired by state-of-the-art approaches to video prediction [31, 8, 20]. To encourage the learning of a disentangled latent representation we minimize total correlation [4] of the latent dimensions and present videos of varying physical properties during training. We train and evaluate our model on synthetic videos of three physical scenarios with varying level of difficulty (sliding objects, collision scenarios). Our experiments demonstrate that our model can learn to disentangle several appearance-related properties such as shape or size of objects. For various dynamics-related physical properties such as speed and friction, we add supervision signals to the latent dimensions and demonstrate that training on total correlation can also improve disentanglement for these properties. To the best of our knowledge, our work is the first to apply total correlation minimization with the aim of discovering physical latent factors in the scene.

The main contributions in this paper are summarized as follows: a) We propose a video prediction model inspired by [31, 8, 20] and train it using total correlation [4]. We also propose an approach to include supervision of dynamics-related properties for representation learning. Our model simultaneously predicts a sequence of future frames and generates latent representations which are physically interpretable for several appearance- and dynamics-related properties. b) We analyze our approach on video datasets of three different physical scenarios with increasing difficulty [1]. We suggest evaluation metrics for reconstruction quality and disentanglement of latent physical properties for the datasets. c) We provide detailed experiments and analysis which demonstrate that our method outperforms several variants in our datasets.

2 Related Work

Learning of Physical Scene Understanding: In recent years, the machine learning community has investigated several approaches to physical scene understanding [33, 22, 32, 31, 28, 25, 30]. Some approaches attempt to learn the dynamics of physical scenes from the explicit state representations (object positions, speed, etc.) which are provided by physics engines [25, 30]. For instance, [25] represents the physical states as a graph and build a learnable and differentiable physics engines to update this graph. The approach in [30] introduces a pipeline to predict the next frame with a physics engine in their structure. Visual interaction networks [28] combine recurrent neural networks and interaction networks [2] to predict the next physical state. Our approach learns state representations and dynamics models directly from video sequences.

More closely related to our approach, instead of utilizing a physics engine to predict the future state, Ye et al. [31] learn to predict the next frame in an end-to-end way. The proposed architecture is an encoder-decoder network which

[1] Dataset available from: https://github.com/TsuTikgiau/DisentPhys4VidPredict

takes four frames in sequence as input and predicts the next frame. For training, the paper proposes a special dataset that consists of multiple small batches where only a single physical property is varied while others are held fixed. Training then imposes a manual assignment of latent variables to physical properties and penalizes deviations of the fixed properties on each batch of sequences from the mean prediction. We instead combine supervision of dynamic properties on specific latent dimensions with a training objective that encourages disentanglement.

Physical Scene Understanding Datasets: A number of studies construct benchmarks with specific properties [29, 24, 21, 23, 31]. For example, Piloto et al. [23] and Riochet et al. [24] focus on the physical plausibility of videos. Lerer et al. introduce a benchmark [21] that includes sequences of wooden-block towers which might collapse for which models need to estimate the trajectories of blocks. [31] contains 5-frame videos of collisions between two objects with simple shapes in a simulator for which the last frame needs to be predicted from the first four frames. Wu et al. [29] record videos in various scenarios in the real world (sliding down a ramp, colliding objects, etc.). The evaluated models need to predict concrete physical properties like bounce height and acceleration. We propose a new video prediction dataset in physical scenes with three scenarios of varying difficulty (sliding objects, colliding objects). In each dataset we vary the physical properties of the objects for which adequate disentangled representations should be learned. Besides image reconstruction metrics, we also propose to use disentanglement metrics.

Video Prediction: Our proposed method is closely related to the field of video prediction. In this field, researchers focus on how to predict a sequence of future frames given a few initial frames [18, 27, 26, 34, 7, 8, 1, 20]. For instance, [26] takes previous frames as input and predicts future frames at the pixel-level. Directly predicting images is prone to loosing details about the appearance of objects though. [27] instead predicts optical flow from the last to the next frame and warps the last frame with the optical flow to generate the prediction. [34] improve the optical flow method using a bilinear sampling layer to make the warping process differentiable. [8] introduces multiple convolutional flow kernels to warp the last frames and composites them into one final output as an alternative to the global optical flow. The optical flow generated images combined with a network stream that directly predicts on the pixel-level. The model also inputs the first frame in the sequence, mostly to maintain information about appearance of objects and background. Dynamics is modelled through LSTMs on the layers of the encoder. Based on [8], [20] applies VAE-GAN [19] for better reconstruction quality. Our network architecture also predicts the future frame in a recurrent VAE structure. We only impose recurrency on the latent state and use total correlation to train for disentanglement.

Representation Learning: Representation learning is an important field to our work. A great deal of previous research has focused on unsupervised representation learning [5, 12, 3, 15, 4]. InfoGAN [5] trains to increase the mutual information between latent codes and generated frames in GANs [10]. Higgins et

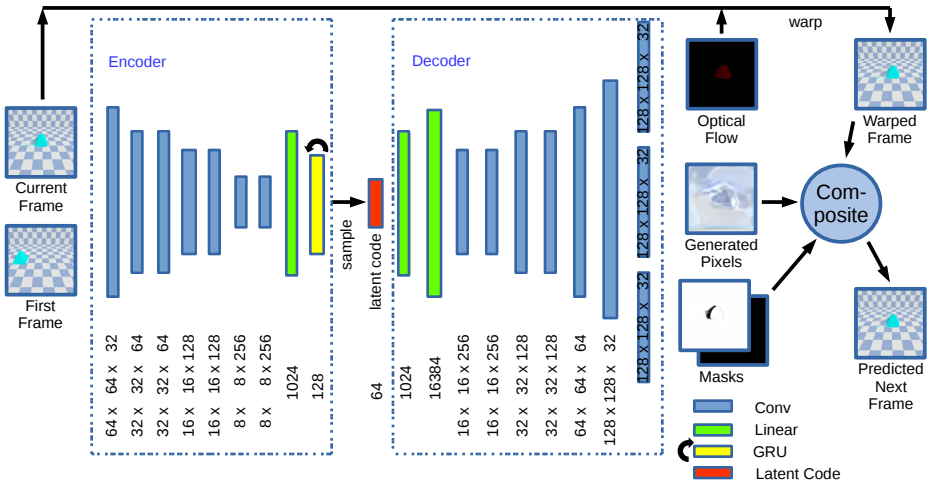


Fig. 1. An overview of our proposed recurrent encoder-decoder network for video prediction. All information about the scene dynamics needs to be maintained in the hidden state of the Gated Recurrent Unit at the last layer of the encoder.

al. [12] analyze that increasing the weight of the KL-divergence loss in VAEs [17] helps to disentangle the latent code. Burgess et al. [3] explain this phenomenon using the information bottleneck theory and propose a method which smoothly decreases the weight for the KL-divergence loss. FactorVAE and β-TCVAE [15, 4] decompose the KL-divergence term into three components and only increase the penalty to the part which is responsible for the disentanglement. Some works also investigate the learning of latent state representations and dynamics models in videos [14, 9, 11]. We apply total correlation minimization to learn a latent state representation to encourage discovering physical latent factors.

3 Method

Our deep learning approach to video prediction uses a recurrent stochastic encoder-decoder architecture which successively predicts the next frame from a sequence of input frames. We train the network using a variational approach which minimizes the total correlation between the encoded latent dimensions. This way, the network is encouraged to learn a representation that disentangles the latent factors of variation in the training videos.

Our model recursively predicts a low-dimensional latent code representation of video sequences. The latent code z_t causally explains the image observations o_t in the video with the observation model $p(o_t \mid z_t)$. For predicting the next latent code, we learn an encoder $q_\theta(z_t|o_{<t})$ that uses information from all previous observations. More specifically, we implement our encoder as a recurrent model. It takes in the previous hidden state s_{t-1} and the last observation o_{t-1} to compute

the next hidden state $s_t = f_\theta(s_{t-1}, o_{t-1})$. The hidden state s_t defines a distribution $\widetilde{z}_t \sim q_\theta(z_t \mid s_t)$ from which the latent code at this step is sampled. The recurrent autoencoder also requires to learn the observation model $p_\psi(o_t \mid z_t)$ with parameters ψ (the decoder).

3.1 Learning Objective

For training this model, we derive a variational lower bound similar to the variational autoencoder [17] and PlaNet [11]. We maximize the data likelihood of the image observations in the video,

$$\begin{aligned} &\ln p(o_{1:T}) = \ln \prod_t \int p(o_t \mid z_t) p(z_t \mid o_{t-1}, s_{t-1}) dz_t \\ &\geq \sum_t \underbrace{E_{q_\theta(z_t \mid o_{t-1}, s_{t-1})} \left[\ln p(o_t \mid z_t)\right]}_{-L_{rec,t}} - \underbrace{KL(q_\theta(z_t \mid o_{t-1}, s_{t-1}) \mid\mid p(z_t \mid o_{t-1}, s_{t-1}))}_{L_{KL,t}}, \end{aligned} \tag{1}$$

where we assume an uninformed Gaussian prior with zero mean and unit diagonal covariance for the state-transition model $p(z_t \mid o_{t-1}, s_{t-1})$. By this approximation, we can use techniques such as β-VAE and β-TCVAE to encourage the latent code to disentangle the latent factors of variation in the training data. The derivation of Eq. 1 can be found in the supplementary material.

The ELBO decomposes in a reconstruction $L_{rec,t}$ and a complexity term $L_{KL,t}$ per time step. We use the Laplace distribution $\frac{1}{2b}\exp(-\frac{|x-\hat{x}|}{b})$ with fixed scale parameter b as the output distribution of decoder. By this, the reconstruction loss can be written as $L_{rec,t} = \frac{1}{b}\sum |x_t - \hat{x}_t|$, where x_t denotes the ground truth frame and $\hat{x}_t$ is the predicted frame. The KL-divergence term can be determined in closed form, since our encoder predicts a normal distribution with diagonal covariance.

The final training objective for our VAE model is

$$L_{VAE} = L_{rec} + L_{KL}, \tag{2}$$

where $L_{rec} = \sum_t L_{rec,t}$ and $L_{KL} = \sum_t L_{KL,t}$.

Recent representation learning approaches have demonstrated that augmentations to this loss function can improve the disentanglement of the representation into the latent factors of variation in the training data. β-VAE increases the penalty to the KL-divergence term,

$$L_{\beta-VAE} = L_{rec} + \beta L_{KL}. \tag{3}$$

Here, $\beta > 1$. β-TCVAE instead decomposes the KL-divergence term into three components

$$\begin{aligned} L_{KL,t} = &\,\mathrm{KL}(q(o_{t-1}, z_t \mid s_{t-1}) \parallel p(o_{t-1} \mid s_{t-1}) q(z_t \mid s_{t-1})) \\ &+ \mathrm{KL}(q(z_t \mid s_{t-1}) \parallel \prod_i q(z_{t,i} \mid s_{t-1})) + \sum_i \mathrm{KL}(q(z_{t,i} \mid s_{t-1}) \parallel p(z_{t,i})) \end{aligned} \tag{4}$$

and only increases the penalty to the total correlation $\mathrm{KL}(q(z_t \mid s_{t-1}) \parallel \prod_i q(z_{t,i} \mid s_{t-1}))$ which is mainly responsible for disentanglement as explained in [4].

Supervision of Latent Dimensions: We also explore training specific dimensions of our latent representation in a supervised way. For selected properties, we normalize their values to the range $[-10, 10]$ and impose an L1 loss between them and specific dimensions of the latent code as an additional loss term. The final training objective in this case is,

$$L_{sup} = L_{unsup} + \lambda \sum_t \sum_i |f_{t,i} - \widetilde{z}_{t,i}|. \tag{5}$$

Here, L_{unsup} is either L_{VAE} or $L_{\beta-VAE}$, $f_{t,i}$ is the value of the i-th property to be supervised at time step t, and $\widetilde{z}_{t,i}$ is the corresponding dimension of the latent code sample.

3.2 Network Structure

Our encoder is a recurrent neural network which receives the last hidden state s_{t-1}, the latest image o_{t-1} and the first image o_1 in the sequence. It outputs a prediction for the state s_t of the next frame which we interpret and split into the mean and diagonal log variances of a normal distribution $q_\theta(z_t|s_t) = q_\theta(z_t|s_{t-1}, o_{t-1})$. The decoder deconvolves samples from the encoder distribution into a Laplace distribution over the pixels in the predicted image. In Fig. 1 we give an overview and details of our network structure.

Besides the last frame, the encoder also takes in the first frame as input for a better conditioning of the reconstruction of background and object shapes. To remember information from previous steps, a GRU [6] layer is used for the last layer of the encoder. Note that the current output of the GRU layer is also its hidden state for the next step (unlike in an LSTM [13]). By this, the model needs to store all information about dynamics in the latent code distribution.

The decoder takes the latent code sample from the encoder and assembles it into the predicted next frame. It first generates a shared feature map via an upsampling network. Then, three small nets convert the shared feature map into optical flow, generated pixels and masks, respectively. The optical flow is used to warp the last frame towards the next frame. Warped frame and generated pixels are composed together via the masks to yield the predicted next frame.

For better image quality, adding skip connections between encoder and decoder or adding recurrency into the decoder are effective approaches [8, 20, 7]. However, these approaches circumvent the representational bottleneck in the latent code and can store dynamics information in other layers. Since we aim at a latent code that represents the appearance and dynamics information required to predict the next frame, we don't adopt these approaches.

Our model takes the first and current frame as input to predict the next frame in each step. The current frame can be either the ground truth data or the predicted one from the last step. At the beginning of the sequence, we feed our model four consecutive ground truth frames to initialize the hidden state

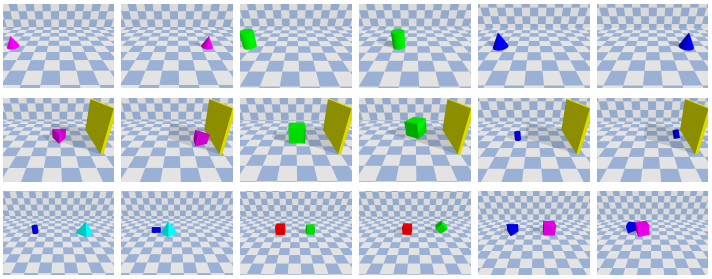


Fig. 2. Samples from our datasets. Row 1, 2, 3 are from the sliding set, the wall set and the collision set, respectively.

of the recurrent encoder. Then, the system recursively uses the predicted image from the last step to perform multi-step prediction.

4 Physical Scene Datasets

We evaluate our approach in videos of physical scenarios of increasing difficulty. We employ the physics engine *PyBullet* to create three datasets. In the sliding set, objects of various shapes and friction coefficients slide with various initial speeds on a plane. The wall set shows collisions of a sliding object with a wall. The collision set contains collision scenarios of two objects that slide into each others. In the latter two, we also vary the density and the restitution coefficients of objects. Example sequences for the datasets are shown in Fig. 2.

For each sequence, we record 10 frames with a rate of 10 Hz. Besides, segmentation masks and depth maps are saved, too. The objects in our dataset have 5 different shapes: cylinder, prism, cube, cone and pyramid. The ratios among edges are fixed, but the scales of objects are changeable for the diversity of data.

Sliding Dataset: The sliding dataset describes a physics scene where an object with various appearances and physical properties slides from left to the right. We do not include sequences in which the object would fall over. Objects in this dataset have 5 properties: shape, scale, friction coefficient, initial speed, and initial position. Different sequences have different combination of these properties which we choose from a finite set of discrete values per property. The set totally has 26000 sequences including a training set with 20000 sequences, a validation set with 3000 sequences, and a test set with 3000 sequences.

Wall Dataset: Similar to the sliding dataset, the objects in the wall dataset also slide from left to the right. However, the object slides into a fixed wall in the right of the scene. If the object is fast enough, it will hit the wall and bounce back. In this dataset, objects may also fall over. We have 5 properties in this set: shape, scale, material, initial speed, and initial position. Each material has its own setting of density, restitution, friction, and color. Again we choose a discrete set of possible values for each property. We have totally 10125 sequences in this

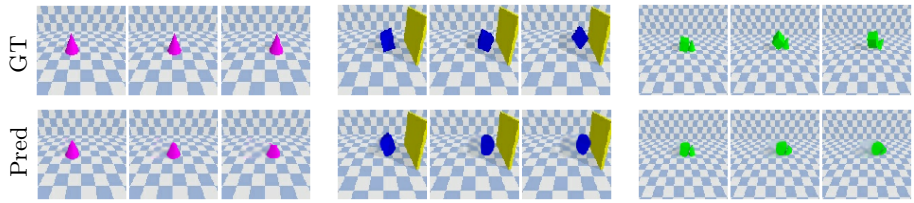


Fig. 3. Prediction examples of β-TCVAE in different datasets. First row demonstrates the ground truth frames. Second row shows the 1st ($t = 5$), 3rd, and 5th predicted frames in three datasets.

set including 7425 sequenes in the training set, 1350 in the validation set and 1350 in the test set.

Collision Dataset: In the collision dataset, 2 objects slide into each other from left and right. Both objects have their own settings of shapes, scales, materials, initial speeds and positions from a discrete set of values. This set has 25000 sequences in the training set, 2500 sequences in the validation set and 2500 sequences in the test set.

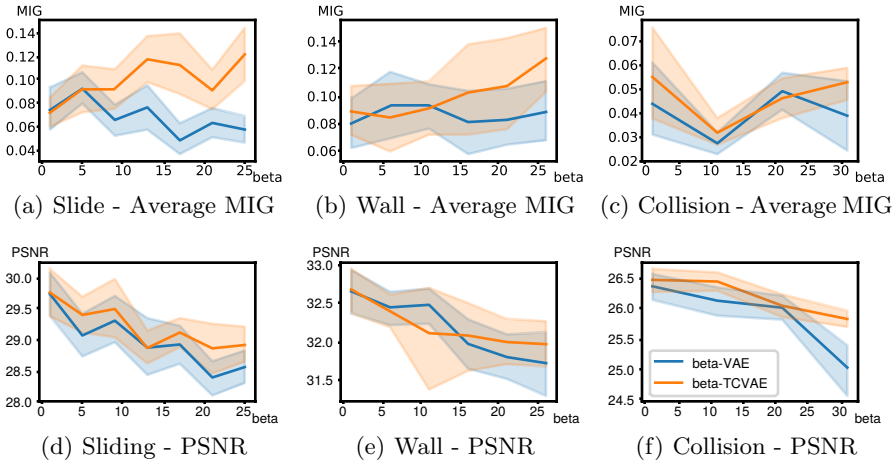
5 Experiments

We evaluate our video representation learning approach on our proposed datasets. To measure the level of the latent code's disentanglement we use the mutual information gap (MIG) proposed in [4]. We also measure the disentanglement of a property separately by computing the mutual information gap for the single property. Additionally, we assess the quality of the video prediction using the peak signal-to-noise ratio (PSNR).

Experiments for Unsupervised Learning: We first assess unsupervised learning with our approach and compare VAE, β-VAE [12] and β-TCVAE [4] objectives for various β values. The models are trained to predict the remaining six frames in each sequence given the first four ground truth frames as inputs.

To explore the relationship between the coefficient β and the level of disentanglement, we evaluate a set of β values. In the sliding set, we set β to 1, 5, 9, 13, 17, 21, 25; in the wall and the collision set, β are set to 1, 6, 11, 16, 21, 26 and 1, 11, 21, 31, respectively. Each setting is trained 22 times in the sliding set and 12 times in the other sets. Each model is trained for 12000 iterations. We use the Adam optimizer [16] with parameters $\beta_1 = 0.5$, $\beta_2 = 0.999$ and learning rate $6e{-}9$. Batch size is set to 8. For the scale parameter of the decoder's Laplace distribution we empirically choose $b = 0.0147$. Schedule sampling [8] is applied for training: The model is first trained to predict only one future step at the beginning of training. Then we smoothly transition to full sequence prediction from iteration 1000 to iteration 9000.

Some prediction examples of β-TCVAE are given in Fig. 3. The average MIG curves are shown in Fig. 4 (a) (b) (c). We show means and 90% confidence intervals of evaluated MIG values. For the sliding set and wall set, a higher β



(a) Slide - Average MIG (b) Wall - Average MIG (c) Collision - Average MIG

(d) Sliding - PSNR (e) Wall - PSNR (f) Collision - PSNR

Fig. 4. MIG and performance reduction (average and 90% confidence intervals) for unsupervised learning. In the sliding set and the wall set, β-TCVAE outperforms β-VAE and successfully increase the average MIG. Besides, larger β leads to bigger performance reduction in both approahces.

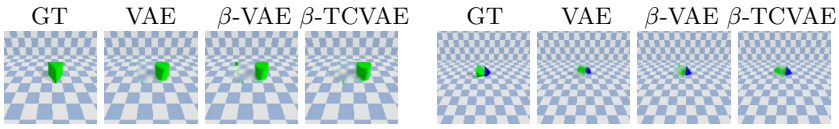
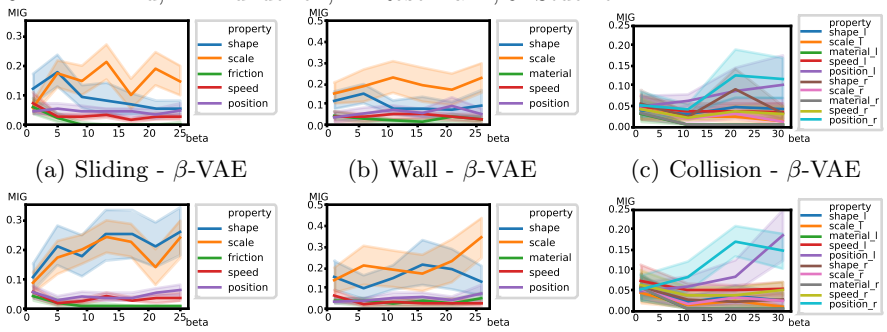


Fig. 5. Predicted last frame of two sequences for different approaches.

helps to increase the average MIG in β-TCVAE. In contrast, β-VAE struggles to improve it. For the most difficult collision set, β-TCVAE slightly improves over β-VAE, while there is no obvious improvement over VAE ($\beta = 1$). Larger β values limit the capacity of the model by forcing it to stay closer to the prior which negatively influences the video prediction quality. This can be seen in Fig. 4 (d) (e) (f) in the reduction in PSNR. We observe that the reduction for β-TCVAE is smaller than for β-VAE in most cases. Fig. 5 shows the last predicted frame in a video sequence by the different approaches.

To figure out which kinds of properties benefit from a larger β, we present the MIGs for individual properties in Fig. 6. Both β-TCVAE and β-VAE can disentangle some properties better like shapes in the sliding set or position in the collision set. β-TCVAE achieves better results than β-VAE. However, the approaches struggle to disentangle dynamic-related properties like speed or friction. To visualize the results of β-TCVAE and β-VAE, we select our best β-TCVAE, β-VAE and VAE ($\beta = 1$) models and show latent traversals for shapes in the sliding set in Fig. 7. For the traversals, we select the dimension of the latent code that has the highest mutual information with the shape.

Experiments for Supervised Learning: Although unsupervised learning in our model using the β-TCVAE objective can improve the disentanglement



(a) Sliding - β-VAE (b) Wall - β-VAE (c) Collision - β-VAE

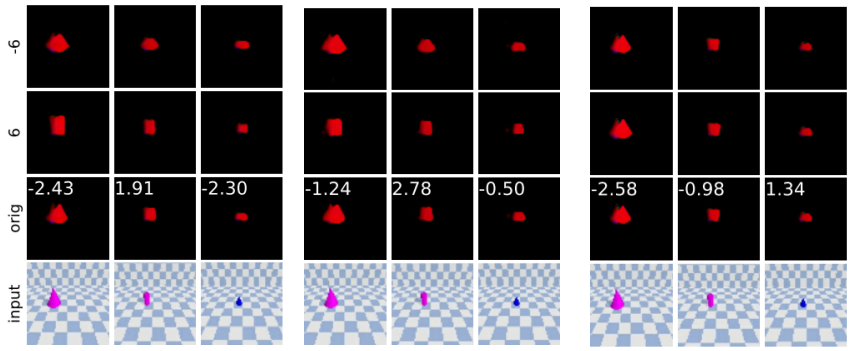
(d) Sliding - β-TCVAE (e) Wall - β-TCVAE (f) Collision - β-TCVAE

Fig. 6. Individual MIG of properties for unsupervised learning. Both β-TCVAE and β-VAE work better for properties which have high influence on the reconstruction loss like shape in the sliding sets and position in the collision set. β-TCVAE outperforms β-VAE in such properties. For other properties they are comparable to VAE ($\beta = 1$).

of properties like shape, scale and position in our datasets, dynamics-related properties like speed and friction are not well disentangled. In this section, we analyze if supervision of some properties can be included into representation learning and if the disentanglement of these properties can be improved.

We train our model in two ways: using the VAE ($\beta = 1$) objective and the β-TCVAE objective with $\beta = 31$. We select dynamics-related properties for supervision in each dataset, and add a supervised loss term for them as detailed in Sec. 3.1. We set $\lambda = \frac{1}{3} \times 10^4$ in our experiments and cap the log variance of the latent code's distribution from below at $\log \sigma^2 = -10$. We train each approach 12 times for 12000 iterations for all datasets. In the sliding set, we supervise friction, speed and position. In the wall and the collision set, speed and position are supervised. The settings of schedule sampling and the Adam optimizer are the same as in the previous experiments.

We show the MIG graphs in Fig. 9. In (a) we observe that β-TCVAE successfully increases the average MIG in all datasets. Subfigures (b) and (d) demonstrate that unlike in the unsupervised learning case, with supervision the model can disentangle dynamic-related properties better in the sliding and collision sets. However, the approach cannot improve the MIG for these properties in the wall set as shown in (c). In addition, the supervised approach also achieves higher MIG for some properties without supervision compared to the unsupervised approach like the shape and the scale in (b). This may be due to the reduction of the supervised properties information in the representation of unsupervised properties. We also show latent traversals in Fig. 10 which compare the results of β-TCVAE and VAE. Fig. 8 demonstrates predictions when dynamics-related properties are changed by their corresponding latent codes in our model. The model trained with the β-TCVAE objective demonstrates a noticable speed change of the objects in this example, making the objects collide. We provide further examples in the supplementary material and video.



(a) β-TCVAE (b) β-VAE (c) VAE

Fig. 7. Latent traversals for the shape property in the sliding set. We manually modify the value (given in row headers) of the dimension corresponding to the specific property and show the predicted optical flows in the first 2 rows. The orig row shows the prediction for the estimated value of the dimension. In the first 2 examples, the contour changes from a triangle-like shape to a rectangular-like shape as we increase the value in β-TCVAE and β-VAE models while this is not the case for standard VAE.

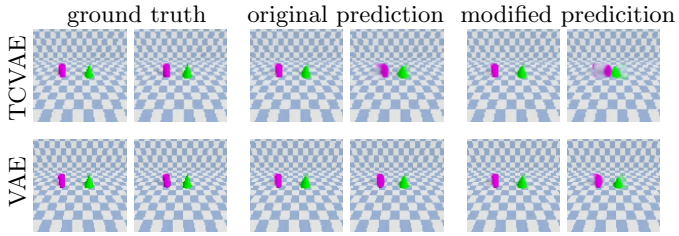
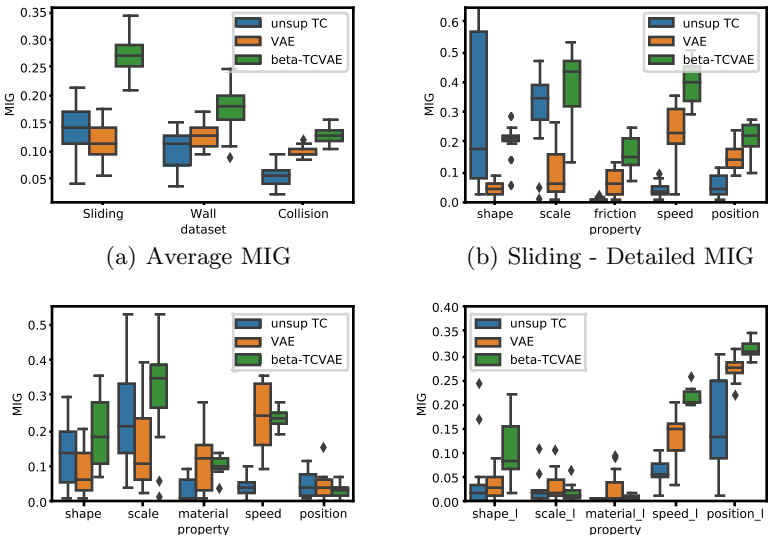


Fig. 8. Effect of latent code modification. We show predicted 1st and last frames from β-TCVAE and VAE. In the last column, we modified the latent code dimension corresponding to the speed at $t = 4$ and show subsequent predicitions. While β-TCVAE generates a collision event, there is only little change for VAE.

6 Conclusion

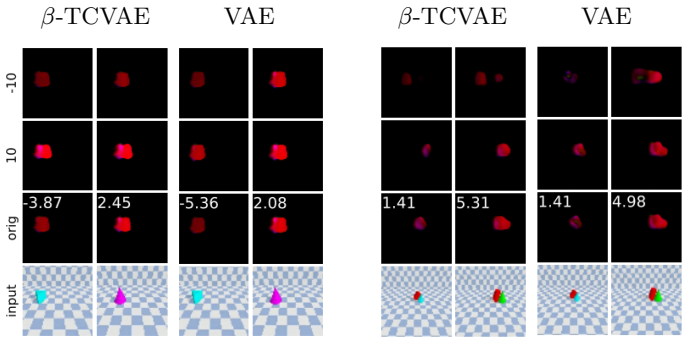
In this paper, we propose a recurrent variational autoencoder model that learns a latent dynamics representation for video prediction. We use total correlation to improve the disentanglement of the learned representation into the latent factors of variation in the training data. In this way, the model can discover several properties related to the physics of the scenarios such as shape or positions of objects. We also demonstrate that partial supervision of dynamics-related properties can be added which further improves the disentanglement of the representation. We evaluate our approach on a new dataset of three physical scenarios with increasing levels of difficulty. In future work we plan to extend our dataset to more complex scenarios and investigate other network architectures to further improve the level of scene understanding.



(a) Average MIG (b) Sliding - Detailed MIG

(c) Wall - Detailed MIG (d) Collision - Detailed MIG

Fig. 9. MIG for supervised learning (mean and 90% confidence intervals) and unsupervised learning for comparison (β-TCVAE). β-TCVAE achieves higher average MIG compared to VAE. There is no significant MIG increase for the wall set for the supervised properties (speed and position). In the sliding and collision sets, β-TCVAE further increases the MIG of the supervised properties (friction, speed and position in sliding set and speed and position in collision set).



(a) Speed (b) Position

Fig. 10. Latent traversals for supervised learning. The brighter red, the faster. β-TCVAE shows more obvious changes of brightness when we modify the corresponding latent code dimension compared to the VAE case. For the property position, the changes for β-TCVAE are also more significant than for VAE.

Acknowledgements

This work has been supported through Cyber Valley.

References

1. Babaeizadeh, M., Finn, C., Erhan, D., Campbell, R., Levine, S.: Stochastic variational video prediction. In: International Conference on Learning Representations (ICLR) (2018)
2. Battaglia, P., Pascanu, R., Lai, M., Rezende, D., Kavukcuoglu, K.: Interaction networks for learning about objects, relations and physics. In: Advances in Neural Information Processing Systems (NIPS) (2016)
3. Burgess, C., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., Lerchner, A.: Understanding disentangling in beta-vae. In: NIPS Workshop on Learning Disentangled Representations (2017)
4. Chen, T., Li, X., Grosse, R., Duvenaud, D.: Isolating sources of disentanglement in vaes. In: Advances in Neural Information Processing Systems (NIPS) (2018)
5. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In: Advances in Neural Information Processing Systems (NIPS) (2016)
6. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. In: NIPS Workshop on Deep Learning and Representation Learning (2014)
7. Ebert, F., Finn, C., Lee, X., Levine, S.: Self-supervised visual planning with temporal skip connections. In: International Conference on Robot Learning (CoRL) (2017)
8. Finn, C., Goodfellow, I., Levine, S.: Unsupervised learning for physical interaction through video prediction. In: Advances in Neural Information Processing Systems (NIPS) (2016)
9. Fraccaro, M., Kamronn, S., Paquet, U., Winther, O.: A disentangled recognition and nonlinear dynamics model for unsupervised learning. In: Advances in Neural Information Processing Systems (NIPS) (2017)
10. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems (NIPS) (2014)
11. Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., Davidson, J.: Learning latent dynamics for planning from pixels. In: Proceedings of the 36th International Conference on Machine Learning (ICML). pp. 2555–2565 (2019)
12. Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., Lerchner, A.: beta-VAE: Learning basic visual concepts with a constrained variational framework. In: International Conference on Learning Representations (ICLR) (2017)
13. Hochreiter, S., Schmidhuber, J.: Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In: Neural computation (1997)
14. Johnson, M., Duvenaud, D.K., Wiltschko, A., Adams, R.P., Datta, S.R.: Composing graphical models with neural networks for structured representations and fast inference. In: Advances in Neural Information Processing Systems 29 (NIPS), pp. 2946–2954 (2016)
15. Kim, H., Mnih, A.: Disentangling by factorising. CoRR **abs/1802.05983** (2018)
16. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (ICLR) (2015)
17. Kingma, D., Welling, M.: Auto-encoding variational bayes (2014)

18. Kitani, K., Ziebart, B., Bagnell, J., Hebert, M.: Activity forecasting. In: European Conference on Computer Vision (ECCV) (2012)
19. Larsen, A., Snderby, S., Larochelle, H., Winther, O.: Autoencoding beyond pixels using a learned similarity metric. In: International Conference on Machine Learning (ICML) (2016)
20. Lee, X., Zhang, R., Ebert, F., Abbeel, P., Finn, C., Levine, S.: Stochastic adversarial video prediction. CoRR **abs/1804.01523** (2018)
21. Lerer, A., Gross, S., Fergus, R.: Learning physical intuition of block towers by example. In: International Conference on Machine Learning (ICML) (2016)
22. Mottaghi, R., Bagherinezhad, H., Rastegari, M., Farhadi, A.: Newtonian scene understanding: Unfolding the dynamics of objects in static images. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
23. Piloto, L., Weinstein, A., T.B., D., Ahuja, A., Mirza, M., Wayne, G., Amos, D., Hung, C.C., Botvinick, M.: Probing physics knowledge using tools from developmental psychology. CoRR **1804.01128** (2018)
24. Riochet, R., Castro, M.Y., Bernard, M., Lerer, A., Fergus, R., Izard, V., Dupoux, E.: IntPhys: A framework and benchmark for visual intuitive physics reasoning. CoRR **abs/1803.07616** (2018)
25. Sanchez-Gonzalez, A., Heess, N., Springenberg, J., Merel, J., Riedmiller, M., Hadsell, R., Battaglia, P.: Graph networks as learnable physics engines for inference and control. In: International Conference on Machine Learning (ICML) (2018)
26. Srivastava, N., Mansimov, E., Salakhutdinov, R.: Unsupervised learning of video representations using LSTMs. In: International Conference on Machine Learning (ICML) (2015)
27. Walker, J., Doersch, C., Gupta, A., Hebert, M.: An uncertain future: Forecasting from variational autoencoders. In: European Conference on Computer Vision (ECCV) (2016)
28. Watters, N., Tacchetti, A., Weber, T., Pascanu, R., Battaglia, P., Zoran, D.: Visual interaction networks. In: Advances in Neural Information Processing Systems (NIPS) (2017)
29. Wu, J., Lim, J.J., Zhang, H., Tenenbaum, J.B., Freeman, W.T.: Physics 101: Learning physical object properties from unlabeled videos. In: British Machine Vision Conference (BMVC) (2016)
30. Wu, J., Lu, E., Kohli, P., Freeman, W., Tenenbaum, J.: Learning to see physics via visual de-animation. In: Advances in Neural Information Processing Systems (NIPS) (2017)
31. Ye, T., Wang, X., Davidson, J., Gupta, A.: Interpretable intuitive physics model. In: European Conference on Computer Vision (ECCV). pp. 89–105 (2018)
32. Zhang, R., Wu, J., Zhang, C., Freeman, W., Tenenbaum, J.: A comparative evaluation of approximate probabilistic simulation and deep neural networks as accounts of human physical scene understanding. In: Annual Conference of the Cognitive Science Society (2016)
33. Zheng, B., Zhao, Y., Yu, J., Ikeuchi, K., Zhu, S.: Scene understanding by reasoning stability and safety. International Journal of Computer Vision (IJCV) **112**(2), 221–238 (2015)
34. Zhou, T., Tulsiani, S., Sun, W., Malik, J., Efros, A.: View synthesis by appearance flow. In: European Conference on Computer Vision (ECCV) (2016)