PhyLoNet: Physically-Constrained long-term Video Prediction *

Nir Ben Zikri[1 [0000−0003−2668−9144]] and Andrei Sharf[1 [0000−0002−3963−4508]]

Ben-Gurion University of the Negev, P.O.B. 653 Beer-Sheva 8410501 Israel
nirey10@gmail.com, asharf@gmail.com

Abstract. Motions in videos are often governed by physical and biological laws such as gravity, collisions, flocking, etc. Accounting for such natural properties is an appealing way to improve realism in future frame video prediction. Nevertheless, the definition and computation of intricate physical and biological properties in motion videos are challenging. In this work, we introduce PhyLoNet, a PhyDNet extension that learns long-term future frame prediction and manipulation. Similar to PhyDNet, our network consists of a two-branch deep architecture that explicitly disentangles physical dynamics from complementary information. It uses a recurrent physical cell (PhyCell) for performing physically-constrained prediction in latent space. In contrast to PhyDNet, PhyLoNet introduces a modified encoder-decoder architecture together with a novel relative flow loss. This enables a longer-term future frame prediction from a small input sequence with higher accuracy and quality. We have carried out extensive experiments, showing the ability of PhyLoNet to outperform PhyDNet on various challenging natural motion datasets such as ball collisions, flocking, and pool games. Ablation studies highlight the importance of our new components. Finally, we show an application of PhyLoNet for video manipulation and editing by a novel class label modification architecture.

Keywords: Deep Learning · Physical Motion · Long-Term Video Prediction.

1 Introduction

Real-world videos often depict the natural motions and dynamics of objects and their interactions. These motions are typically governed by the physical and biological laws of nature. Accounting for such natural properties is an appealing way to improve realism in future frame video prediction. Nevertheless, the definition and computation of intricate physical and biological properties in motion videos are challenging. Thus, a key problem is to design video prediction methods able to represent the complex dynamics underlying raw data.

* This research was partially supported by the Lynn and William Frankel Center for Computer Science at BGU.

Video forecasting consists of predicting the future content of a video conditioned on previous frames. In this context, a key problem is to design video prediction methods able to represent the complex dynamics underlying raw data. In this work, we focus on unsupervised video prediction of complex motions and interactions, typically governed by physical or biological laws. Our datasets lack of semantic labeling forces the network towards predicting motions only from domain knowledge in an unsupervised manner.

We introduce PhyLoNet, a PhyDNet [21] extension that learns long-term future frame prediction of natural motions and their manipulation. In our experiments, we explore natural motions and interactions such as bird flocking motions, multiple ball collisions, and different movement behaviors. We model the motion dynamics by incorporating our deep learning framework with PDE modules and a motion-aware loss. Thus, we account for physical, biological, and in general high-order priors in the motion video. Essentially, our method builds upon the two-branch PhyDNet architecture which allows disentangling the physical properties of a motion video from other factors in a latent space. We introduce a novel relative flow loss (denoted RF-loss) which together with a customized deep network architecture yields a significant improvement in future frame prediction of complex motions and interactions. We also extend our neural network design to accomplish future frame manipulation besides the classical prediction task. Thus, we incorporate motion targets and paths as additional class labels and channels in the network architecture. We show that at test time, it is possible to adjust class labels on the fly, by doing so we allow editing and controlling of the video prediction sequence. In our experiments, we show that our model is able to predict future frames with significantly higher accuracy than PhyDNet and produce state-of-the-art results. We also show an application of our framework for video completion and editing. To summarize our work makes the following contributions:

- We introduce a novel relative flow loss (RF-loss) that accounts for the complex motion dynamics in the video and significantly improves prediction accuracy and quality.
- We introduce PhyLoNet architecture which modifies PhyDNet architecture for improved long-term predictions and domain generalization.
- We extend PhyLoNet to allow video editing through on-the-fly prediction manipulation and control.

2 Related Work

Predicting the next frame of a video has received growing interest in the computer vision community over the past few years. Recent works have managed to achieve state-of-the-art performances using deep neural networks for next-frame video prediction tasks. Sequence to sequence LSTM and Convolutional variants [42, 44] are the core of many similar studies [9, 59]. Further works explore various Recurrent Neural Network (RNNs) [31, 52–55] and 2D/3D ConvNets [4, 28, 39, 51] architectures.

This task becomes even more challenging when facing high-dimensional images, therefore predicting the geometric transformations between frames [3,9,60] or using their optical flow [22–24, 27, 34] reduce substantially the complexity of image generation. This approach is usually effective for single frame prediction [11], where it aims to predict the future frame according to the whole information within a frame as one representation, but it fails when predicting frames for the long-term. Another approach for handling high-dimensional inputs is by disentangling independent factors of variations in order to apply the prediction to lower-dimensional representations [8, 18, 55].

Relational reasoning which is often implemented with graphs [2,17,32,40,45] accounts for basic physical laws, e.g. drift, gravity, spring [30, 56, 57]. Still, these methods fail for general real-world video forecasting.

A promising line of work focuses on disentangling approach which factorizes the video into independent components [6, 10, 13, 48, 50]. Some works disentangles content and motion [13, 50, 58] while others disentangle deterministic and stochastic factors [6]. [51] propose a generative adversarial network for the future frame prediction based on foreground-background mask disentanglement. Another foreground-background disentanglement approach is to use segmentation masks [29] extracted from a semantic segmentation network and use it to increase the attention over each instance individually [58].

Physics and PDEs. Exploiting prior physical knowledge is another appealing way to improve prediction models. Solving PDEs with DNNs [35, 36, 41] has grown a lot of attention in recent years, more specifically, a connection between PDEs and CNN's [25, 26] shows that it is possible to learn filters that resemble set of differential orders, combining them with temporal based DNN's like LSTM [43] significantly improves physical dynamics prediction in latent space.

Some works [21, 33, 61] find physical models insufficient and propose to combine physical models with data-driven models in order to achieve all the data within a frame. [1] approach uses multiple physical models in order to extract different physical properties, those models are controlled by a Transformers [7, 49] network which implements the Mixture of Experts concept for selecting the best physical models that represent the video dynamics the most.

Leveraging physical knowledge is not enough for general video forecasting. Despite the fact that it can learn a broad class of PDEs, there are a lot of domains where this physical prior might not fit.

Dedicated Loss. Dedicated loss functions [5, 12] and the ability of Generative Adversarial Networks to generate high-quality frames [20, 28, 51] have been investigated. However, combining GANs with prior information, such as physical models, remains an open topic for research.

In this work, we design a novel motion loss that is based on optical flow and denoted as *Relative Flow Loss* (RF-loss). The main advantage of deep optical flow is its differentiability trait, it allows utilizing it in deep neural networks as a loss that relates directly to the object's speed and direction. In our context, we use our RF-Loss to track and differentiate an object's motion distortion in an unsupervised manner.

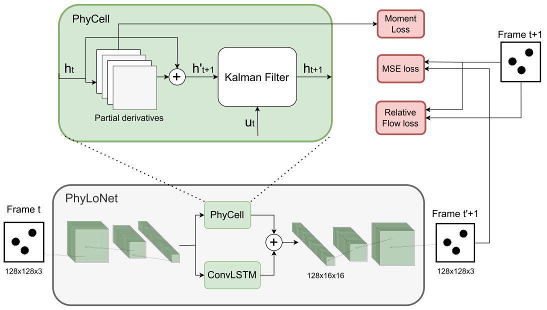

Fig. 1: The PhyLoNet architecture. An image frame is first encoded into a latent vector h_t, transferred into the PhyCell and ConvLSTM cells whose outputs are summed and decoded into the next frame.

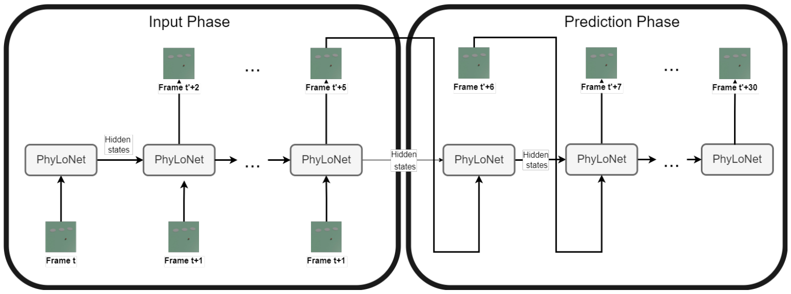

Fig. 2: The unrolled PhyLoNet architecture.

Recent works show that it is possible to compute the optical flow using various DNN architectures [14,15,37,46]. These are mostly based on pyramid structures that learn different order filters which enable to capture of different receptive fields. Another approach is based on recurrent units and per-pixel features [47].

3 Technical Details

In this work, we focus on learning long-term motion video prediction and manipulation of future frames. Our PhyLoNet network is based on PhyDNet [21]. The network consists of a two-branch deep architecture that explicitly disentangles physical dynamics from unknown complementary information (see figure 1). One branch is built from ConvLSTM cell in order to retain the residual information within a video. The second branch is built from a Physical cell (PhyCell) that aims to learn the physical dynamics for long-term predictions using the deep Kalman mechanism. In contrast to [21] we customize the encoder-decoder architecture and remove skip connections in order to allow more accurate and longer predictions. Similar to [21] we use an image moment loss and MSE loss between the ground-truth and the model's prediction. The third loss is our novel relative flow loss which also allows more accurate and longer predictions.

3.1 PhyLoNet network

PhyLoNet is trained over raw video frames in an unsupervised way. In our case, we experiment with motion videos of bouncing and colliding balls, bird flocking, and a pool game. The network trains in an unsupervised manner on raw motion videos of a specific motion domain without any semantic knowledge or labels.

Figure 2 presents an overview of the unrolled architecture of our network. The PhyLoNet architecture is described in figure 1. The physical cell learns the PDEs underlying our dynamics domain where each filter approximates PDE coefficients of different orders. Image moments are used to better approximate the PDE coefficients. Both prediction $h_{t+1}^{'}$ and observation u_t (the encoded ground truth frame) are inserted into a deep Kalman filter [19] for both estimation improvement and long-term predictions.

The encoder-decoder architecture in PhyLoNet is different than PhyDNet. Instead of using a U-Net based encoder-decoder [38] which relies heavily on the skip-connections for generating future frames, we use simple CNN based encoder-decoder [62] without any skip connections. This approach allows a generation with more flexibility from the latent vector.

Nevertheless, removing the skip connections might result in poor frame quality. In order to preserve the perceptual information in the video and increase the frame quality, we use the perceptual loss as presented in [16]. The perceptual loss is based on a pre-trained VGG-16 network and constructed as follows:

$$L_{perceptual} = \alpha \cdot L_{content} + \beta \cdot L_{style} \tag{1}$$

$L_{perceptual}$ is the perceptual loss, $L_{content}$ and L_{style} are the content and style loss controlled by the coefficients α and β respectively.

$$L_{content} = \sum_{l \in F_c} ||P_l(G) - P_l(P)||_2^2 \tag{2}$$

$$L_{style} = \sum_{l \in F_s} ||\psi_l(G) - \psi_l(P)||_F^2 \tag{3}$$

P_l defines the feature vector of layer l extracted from VGG-16 network, G is the ground-truth frame and P is the prediction frame. ψ_l is the Gram matrix of feature vector l and F stands for Frobenius distance. F_s and F_c are the style and content layers from the VGG-16 network. The intuition behind the perceptual loss is to extract the style and content features from a pre-trained network instead of using hand-crafted labels, by doing so, the training process remains unsupervised.

3.2 Relative Flow Loss

Let $[g_1, g_2, ...g_N]$ be the ground truth frame sequence and Let $[p_1, p_2, ...p_N]$ be the predicted frames sequence (see Figure 3). OF is the optical flow, which is

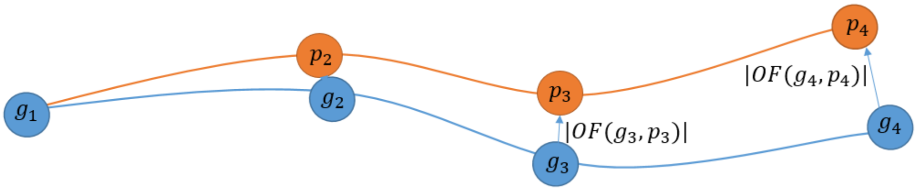

Fig. 3: The relative flow loss (RF-Loss) illustration. The loss measures the flow between the ground truth motion frames (g2-g4) and the predicted motion frames (p2-p4).

the pattern of apparent motion. The optical flow is a 2D vector field where each vector is a displacement vector showing the movement of points from the first frame to the second. The relative flow loss is defined as the optical flow between corresponding ground truth and predicted frames along the video sequence. This is an effective loss in the unsupervised learning process since frames lack any semantic annotations and labels.

In the traditional optical flow loss (OF-loss), the optical flow is calculated between two consecutive frames:

$$L_{of} = \sum_{t=1}^{N-1} ||OF(g_t, g_{t+1}) - OF(p_t, p_{t+1})||^2 \tag{4}$$

In contrast, RF-loss is calculated as follows:

$$L_{rf} = \sum_{t=1}^{N} |OF(g_t, p_t)| \tag{5}$$

where the optical flow is calculated by the RAFT network [47] and N is the number of predicted frames.

The motivation behind the relative flow loss is to increase the penalty for frames with objects that deviate from their original route at further time steps. Standard optical flow loss is focused locally on two consecutive images and measures the directional differences and deviations. Instead, the relative flow loss measures the prediction error between pairs of ground truth and the corresponding predicted frame. Thus, instead of measuring pixel-level differences locally, our loss accounts for global deviations and motions between ground truth and predicted frames. Figure 3 illustrates the RF-loss calculation. Blue/orange balls refer to corresponding (w.r.t. time step) GT/predicted frames respectively. We calculate at each time step the optical flow between ground truth and prediction, then we take the absolute value as the RF-loss, which resembles the global motion structural error between the frames.

3.3 Video Manipulation

Video manipulation refers to the ability to influence the predicted motion inside the video. To achieve this, we allow our network to obtain class labels along

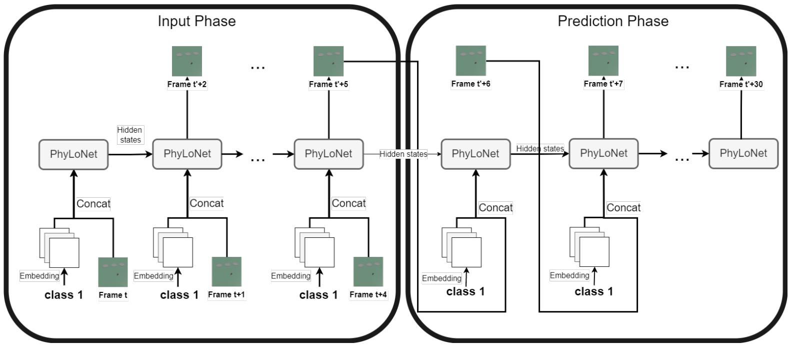

Fig. 4: The label constrained PhyLoNet. The **class 1** label is fed for video prediction editing.

the prediction process that are concatenated to the frames. See Figure 4 for the network architecture.

Manipulating the video toward the desired class requires assimilating a label into the frame representation. LSTMs do not accept all frames at once, instead, it performs an iterative feeding of the previous states while keeping a ”memory” of the important parts of each hidden state. Therefore, in order to create substantial assimilation of the label, we embed the class label in the same shape as the image and concatenate it to the image. The encoder then accounts for both the label and the image when encoding them together into the latent space.

In Figure 4 we illustrate the PhyLoNet architecture for video manipulation. First, we embed our desired class label and reshape it into our image dimension, then we perform the concatenation and forward it to the PhyLoNet for the next frame prediction. The embeddings are for both the input and prediction phases.

3.4 Loss Functions

Figure 1 presents in red blocks the loss functions at time step t. The total loss function of our network is defined as

$$L_{total} = \gamma \cdot L_{moment} + \delta \cdot L_{perceptual} + \epsilon \cdot L_{rf} \tag{6}$$

Where L_{moment} is the image moment loss, $L_{perceptual}$ is the perceptual loss, and L_{rf} is our proposed relative flow loss. We set γ, δ, ϵ to be 1, 1 and 0.00005 respectively. The coefficients α and β in $L_{Perceptual}$ are set to 0.1 and 0.05.

4 Results

We have run several video prediction and manipulation experiments, comparisons, and ablation studies to evaluate our network. To generate a ground truth, we have generated motion video datasets using various simulations. Then, we trained our network for each motion video dataset on an RTX-2080 GPU and predicted future frames. In the video manipulation experiments, we trained on a GTX-1080 GPU. All models in experiments are trained with a batch size of 4 and image size of 128x128 except the experiment of **Sin-Moving-Ball-Labeled** dataset which has a batch size of 8 and image size of 64x64. We used an Adam optimizer with a learning rate of $1e-4$. Our code is available at https://github.com/nirey10/PhyLoNet.

4.1 Datasets

We have generated the following synthetic datasets for our experiments.

Ball-Collisions - The data is generated by the Unity physics engine. Each video consists of 3 balls at random positions within a solid bounding frame. Each ball is initialized with a random speed vector. Balls are moving upon a fraction-less surface, which means no energy loss. This Dataset consists of 1800 training videos and 200 test videos. Each video is composed of 30 frames.

Moving-Ball-Labeled - This dataset consists of labeled videos of 3 categories, go-left, go-middle, and go-right. Each category consists of 300 videos of 30 frames each. Videos in each category depict random routes towards the desired target from different camera angles and for 3 different balls.

Sin-Moving-Ball-Labeled - This dataset is similar to the **Moving-Ball-Labeled** dataset with one main difference regarding the ball motion. Specifically, we introduce specific motion characteristics for each label. In the go-left, balls have a straight movement behavior, in the go-middle label, balls have a small sinus amplitude in their movement, and in the go-right label balls have a big sinus amplitude movement behavior.

Flocking - The flocking dataset consists of a simulation of 30 birds moving around as a flock based on swarm optimization. Flocking is simulated using the Pygame library. This dataset contains 1800 videos for training and 200 for test. Each video is composed of 100 frames.

Pool - The Pool dataset consists of a simulated pool game. The data is generated by the Unity engine. It consists of 2000 videos of 30 frames each separated into 2 classes. Class 0 represents ”miss” and class 1 represent ”score”. The black ball is the cue ball and the blue ball (target) is placed randomly in front of the cue ball. The pot is an orange rectangle surrounded by walls on each side. The videos consist of various ball initializations in term of position, speed, and heading degree.

4.2 Ball Collisions

The ball collision dataset demonstrates physical dynamics and collision interactions between balls, surrounded by solid frames, and other balls. In this experiment, the system obtains 5 frames as input and then predicts the next 100 frames.

Figure 5 shows prediction comparisons between PhyDNet (top) and PhyLoNet (bottom). Figures show several predicted frames overlayed on top of each

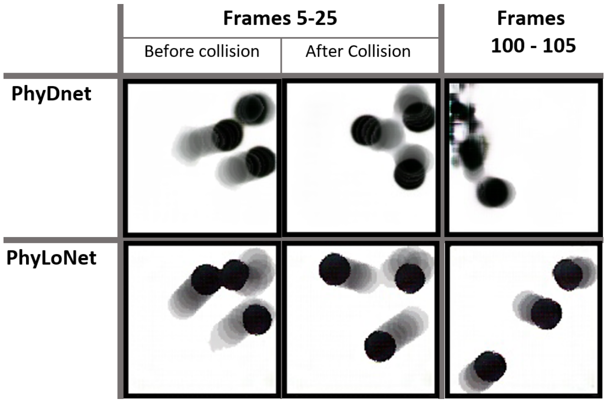

Fig. 5: Three balls collision prediction comparison. Frames 5-25 demonstrate short-term ball motion before and after the collision, while frames 100-105 show long-term prediction.

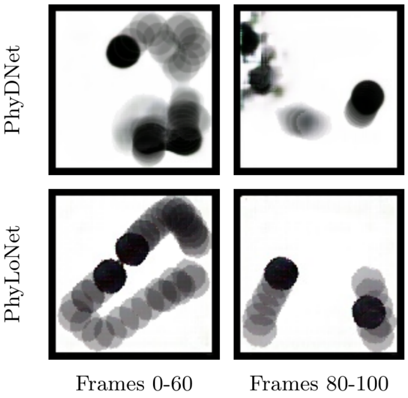

Fig. 6: Demonstration of the PhyDNet and PhyLoNet model's generalization over 2 balls collision.

other with opacity for demonstration of motion prediction. Given the initial 5 input frames, both models are able to predict ball collisions at frames 5-25. A closer look will show that the physical dynamics are much more realistic in our method, see the motion angle of 2 topmost balls after the collision, as well as the ball shape conservation. Frames 100-105 show the quality differences of our long-term prediction compared to PhyDNet, we preserve both physical dynamics and residual information intact.

4.3 Ball Collisions Generalization

We test the generalization capability of our network. For this purpose, we create a dataset consisting of 2 ball collision scenes and evaluate its prediction quality over our pre-trained network that was trained on the 3 balls dataset (Ball-Collisions dataset).

Figure 6 compares generalization of both PhyDNet and PhyLoNet on 2 balls motion over 100 frames. As can be seen, PhyDNet generates frames with 3 balls at the very first predictions and is unable to generalize for 2 balls while our network is able to produce high-quality and realistic results in terms of

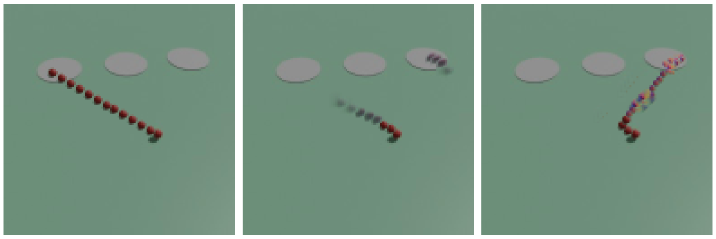

Fig. 7: Ball movement manipulation from ground truth route (left image) towards the go-right target using PhyDNet (middle image) and our method (right image).

both collision dynamics and the structural preservation. Frames 80-100 show that our generalization holds for long-term prediction as PhyLoNet predicts realistic and high-quality ball motions while PhyDNet contains severe artifacts and inadequate dynamics.

4.4 Video Manipulation

In order to demonstrate video manipulation, we experiment with the **Moving-Ball-Labeled** and **Sin-Moving-Ball-Labeled** datasets. Specifically, ball motions consist of additional class labels according to the video content. Thus, given an initial input motion sequence that relates to one of the classes, the network aims at predicting future frames according to a different class label chosen by the user while maintaining the consecutive path of the moving objects.

In Figure 7 the ground truth ball motion was go-left target. Taking only the first 5 frames of the motion, PhyDNet and PhyLoNet were given also a go-right label, in order to manipulate the network prediction.

PhyDNet was unable to manipulate the ball movement towards the right target, instead, the ball disappears in-between and comes back near the right circle in the last frames. In contrast, our network predicted a continuous ball movement towards the right target. Artifacts are due to a lack of residual connections which reduce output quality but allow freedom in the frame generation. In addition, the sharp trajectory changes after 5 input frames indicate the powerful control of our model.

Similarly, we demonstrate the ability to manipulate ball movement also w.r.t. their style and not only their target. We use our **Sinus-ball-Movement-Labeled** dataset in order to train on three different movement class labels: straight, small sinus amplitude, and big sinus amplitude movements.

In Figure 8 the bold images in the diagonal are the ground-truth videos before manipulation. Thus, we show three different samples where each sample has a different target and style.

We then take each ground truth sample and manipulate it to a different target (rows). Our model is able to augment the original route towards the new destination label together with the corresponding movement style w.r.t. sinus amplitude getting bigger as we move to the ”right” label.

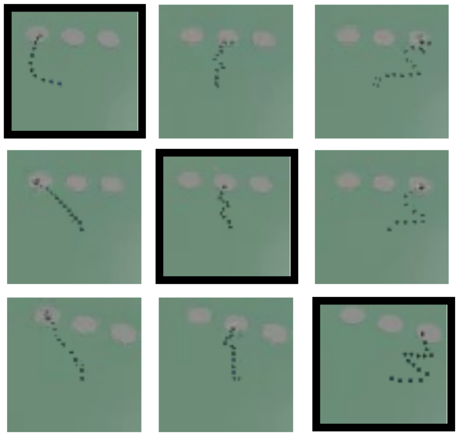

Fig. 8: PhyLoNet video manipulation of motion target and style. Each row shows a different ground-truth label (black framed images) and its manipulation of the other class labels.

4.5 Flocking

In this experiment, we test video prediction, in pixel space, of flocking motions generated by a swarm optimization. We train our network on the **Flocking** dataset. In the prediction step, input for our models is solely 5 frames and we predict the next 100 frames.

Figure 9 shows the flocking ground truth motion (left col) and a comparison between PhyDNet (mid-col) and our network (right-col) flocking motion prediction. The top row is the merged frames and the bottom row is the predicted flocking motions emphasized by colored vectors.

PhyDnet was unable to predict accurate flocking behavior such as cohesion. The green and red motions are performing cohesion in GT but they do not continue with the average direction in the prediction. Similarly, the blue motion is interfered by other birds and does not continue toward the dedicated target.

In contrast, our model predicts accurate cohesions for all 3 cases in this scene while keeping the speed and average heading intact.

4.6 Pool

We present another video manipulation experiment on our **Pool** dataset. The data consists of pool ball interactions labeled with ”score” and ”miss” categories. We then feed the model with 5 frames as input and predict the next 25 frames according to the desired label category as given by the user.

Figure 10 demonstrates manipulation results for a pool ball after its collision with the cue ball. The orange rectangle is the ”pot” surrounded by 2 solid walls on each side. Left-to-right, starting from an initial setup the blue ball does not move until the collision occurs, the initial cue ball’s movement is demonstrated using an arrow. The original GT ball ”miss” movement is shown (mid-left) followed by PhyDNet (mid-right) and PhyLoNet (right) ”score” manipulation.

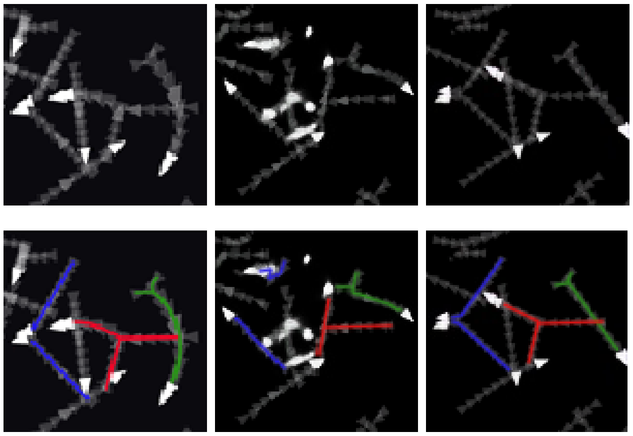

Fig. 9: Flocking motion prediction. Left-column is the GT motion, PhyDNet prediction is middle-column and our PhyLoNet prediction is right-column.

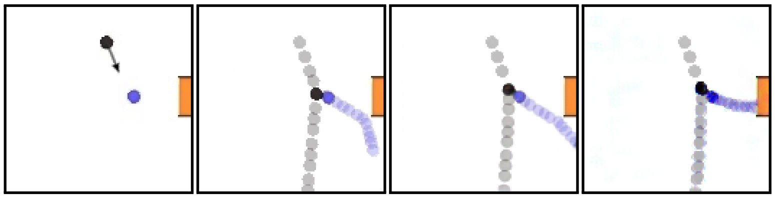

Fig. 10: Demonstration of a ball's route change towards the class label "score". The original class label of the sample is "miss".

Our model is capable of manipulating the original scene towards our desired label, it augments the ball's route directly to the orange "pot" and produces a "score" labeled sample. The PhyDNet results are very similar to the ground truth and the model is unable to manipulate the video toward our desired class.

4.7 Moving-MNIST Comparison

Table 1 presents quantitative results of PhyLoNet compared to other baseline methods on the Moving-MNIST dataset. Although PhyLoNet was able to achieve better results from most of the baselines it just arrived second to PhyDNet. There are two major reasons for that. First, Moving-MNIST is not a dataset suited for our network. I.e, our method focuses on motions with physical properties, which is not the case here. Secondly, the metrics applied, MSE, MAE, and SSIM, do not reflect the performances on future physical interactions (see chapter 4.8).

Method	MSE	MAE	SSIM
ConvLSTM	103.3	182.9	0.707
PredRNN	56.8	126.1	0.867
Causal LSTM	46.5	106.8	0.898
MIM	44.2	101.1	0.910
E3D-LSTM	41.3	86.4	0.920
PhyDNet	**24.4**	**70.3**	**0.947**
PhyLoNet	34.5	93.5	0.921

Table 1: Quantitative results of PhyLoNet compared to baseline models using Moving-MNIST dataset.

4.8 Ablation Study

We show the effect of our network architecture and our proposed loss in the ablation study experiments. We refer to PhyDNet as the original architecture [21] and PhyLoNet as our proposed architecture. The loss functions we incorporate are the Relative Flow loss (RF-Loss) and the original Optical Flow loss as described in section 4.2.

Relative Flow Loss For Complex Dynamics Prediction. We evaluate the effect of our relative flow loss compared to the standard optical flow approach and the original PhyDNet model. We perform an ablation study on the 3-ball collision dataset. Since small deviations in collision returning angles cause large deviations over time, we compare only the first 30 frames after the collision.

In Figure 11, we compare the performances of PhyDNet, PhyLoNet with RF-Loss, and PhyLoNet with Optical Flow Loss models. The left graph shows MSE error with GT. We observe that the MSE method is quite general metric and does not reflect the performances on future physical interactions, especially for multi-object collisions. In order to perform a better, customized evaluation of our method performance we define a new metric that tracks each ball position over time. The object tracking is implemented using OpenCV's CSRT object tracker. We compare the cumulative Euclidean distances of every ball centroid between the predicted and ground-truth frames. In Figure 11 (right) we can see the model's comparison using our multi-object tracking metric.

In all metrics PhyLoNet outperforms PhyDNet, indicating the strong effect of the encoder-decoder over long-term prediction. The RF-Loss also shows a significant contribution to the model performance. Last, we see that RF loss outperforms the optical flow loss for the prediction of complex physic dynamics.

Relative Flow Loss For Video Manipulation. In this study, we evaluate the effect of our relative flow loss on the **Pool** dataset which combines both physical interactions, collisions, and video manipulation. In this context, we evaluate the balance between accurate physical dynamics prediction and the flexibility to perform manipulations.

Figure 12 presents two different manipulation examples. Each example is originally labeled as a "miss" and the red arrow marks the ground-truth motion vector of the blue ball after the collision. The left image in each example shows the "score" manipulations using PhyLoNet without RF-Loss, while the right

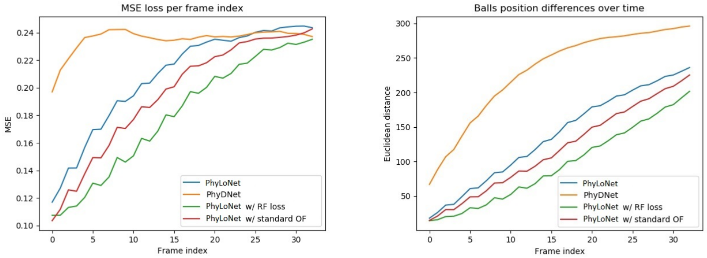

Fig. 11: Ablation study results for the **Ball-Collision-3** dataset using two different metrics

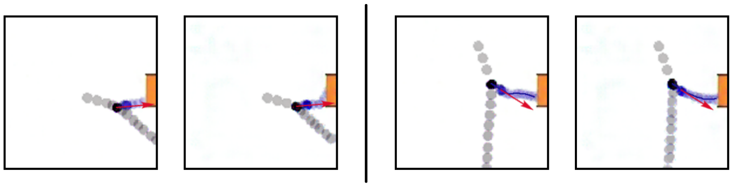

Fig. 12: Two examples (separated by a line) of manipulating the ”miss” label into a ”score” label using two different models: PhyLoNet without RF-Loss (left image) vs. PhyLoNet with RF-Loss (right image).

image shows PhyLoNet with RF-Loss. Both models are able to manipulate the video towards the ”score” label. Nevertheless, PhyLoNet with RF-Loss is more faithful to the GT data and therefore the ball motion is more restricted by the original dynamics.

5 Conclusions and Limitations

We introduce an enhanced PhyDNet design for unsupervised long-term video dynamics prediction, the PhyLoNet. We introduce a novel loss that accounts for complex dynamics. We also introduce a model to control and manipulate videos by changing their class label. The results demonstrate that our model is capable of handling complicated physical dynamics for long-term prediction. We believe this work is the basis of further physically constrained video prediction tasks and its contributions can be applied to more complicated motion domains.

In terms of limitations, we have tested our models only on synthetic data with a smooth static background. This is because we aimed at focusing on the physical and biological dynamics instead of dealing with computer vision tasks such as FG/BG separation, background stabilization and etc. More complex video settings and in general in-the-wild datasets are currently left for future work.

References

1. Aoyagi, Y., Murata, N., Sakaino, H.: Spatio-temporal predictive network for videos with physical properties. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 2268–2278 (2021). https://doi.org/10.1109/CVPRW53098.2021.00256
2. Battaglia, P.W., Pascanu, R., Lai, M., Rezende, D., Kavukcuoglu, K.: Interaction networks for learning about objects, relations and physics (2016)
3. Brabandere, B.D., Jia, X., Tuytelaars, T., Gool, L.V.: Dynamic filter networks (2016)
4. Byeon, W., Wang, Q., Srivastava, R.K., Koumoutsakos, P.: Contextvp: Fully context-aware video prediction (2017). https://doi.org/10.48550/ARXIV.1710.08518, https://arxiv.org/abs/1710.08518
5. Cuturi, M., Blondel, M.: Soft-dtw: a differentiable loss function for time-series (2017)
6. Denton, E., Birodkar, V.: Unsupervised learning of disentangled representations from video (2017)
7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale (2021)
8. Eslami, S.M.A., Heess, N., Weber, T., Tassa, Y., Szepesvari, D., Kavukcuoglu, K., Hinton, G.E.: Attend, infer, repeat: Fast scene understanding with generative models (2016)
9. Finn, C., Goodfellow, I., Levine, S.: Unsupervised learning for physical interaction through video prediction (2016). https://doi.org/10.48550/ARXIV.1605.07157, https://arxiv.org/abs/1605.07157
10. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning (2015)
11. Gao, H., Xu, H., Cai, Q.Z., Wang, R., Yu, F., Darrell, T.: Disentangling propagation and generation for video prediction (2019)
12. Guen, V.L., Thome, N.: Shape and time distortion loss for training deep time series forecasting models (2019)
13. Hsieh, J.T., Liu, B., Huang, D.A., Fei-Fei, L., Niebles, J.C.: Learning to decompose and disentangle representations for video prediction (2018)
14. Hui, T.W., Tang, X., Loy, C.C.: Liteflownet: A lightweight convolutional neural network for optical flow estimation (2018)
15. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: Flownet 2.0: Evolution of optical flow estimation with deep networks (2016)
16. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution (2016)
17. Kipf, T., Fetaya, E., Wang, K.C., Welling, M., Zemel, R.: Neural relational inference for interacting systems (2018)
18. Kosiorek, A.R., Kim, H., Posner, I., Teh, Y.W.: Sequential attend, infer, repeat: Generative modelling of moving objects (2018)
19. Krishnan, R.G., Shalit, U., Sontag, D.: Deep kalman filters (2015)
20. Kwon, Y.H., Park, M.G.: Predicting future frames using retrospective cycle gan. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1811–1820 (2019). https://doi.org/10.1109/CVPR.2019.00191
21. Le Guen, V., Thome, N.: Disentangling physical dynamics from unknown factors for unsupervised video prediction. In: Computer Vision and Pattern Recognition (CVPR) (2020)

22. Li, Y., Fang, C., Yang, J., Wang, Z., Lu, X., Yang, M.H.: Flow-grounded spatial-temporal video prediction from still images (2018)
23. Liang, X., Lee, L., Dai, W., Xing, E.P.: Dual motion gan for future-flow embedded video prediction (2017)
24. Liu, Z., Yeh, R.A., Tang, X., Liu, Y., Agarwala, A.: Video frame synthesis using deep voxel flow (2017)
25. Long, Z., Lu, Y., Dong, B.: Pde-net 2.0: Learning pdes from data with a numeric-symbolic hybrid deep network. Journal of Computational Physics **399**, 108925 (Dec 2019). https://doi.org/10.1016/j.jcp.2019.108925, http://dx.doi.org/10.1016/j.jcp.2019.108925
26. Long, Z., Lu, Y., Ma, X., Dong, B.: Pde-net: Learning pdes from data (2018)
27. Luo, Z., Peng, B., Huang, D.A., Alahi, A., Fei-Fei, L.: Unsupervised learning of long-term motion dynamics for videos (2017)
28. Mathieu, M., Couprie, C., LeCun, Y.: Deep multi-scale video prediction beyond mean square error (2015)
29. Mo, S., Cho, M., Shin, J.: Instagan: Instance-aware image-to-image translation (2019)
30. Mrowca, D., Zhuang, C., Wang, E., Haber, N., Fei-Fei, L., Tenenbaum, J.B., Yamins, D.L.K.: Flexible neural representation for physics prediction (2018)
31. Oliu, M., Selva, J., Escalera, S.: Folded recurrent neural networks for future video prediction (2017)
32. Palm, R.B., Paquet, U., Winther, O.: Recurrent relational networks (2017)
33. Pan, T., Jiang, Z., Han, J., Wen, S., Men, A., Wang, H.: Taylor saves for later: disentanglement for video prediction using taylor representation (2021)
34. Patraucean, V., Handa, A., Cipolla, R.: Spatio-temporal video autoencoder with differentiable memory (2015)
35. Raissi, M.: Deep hidden physics models: Deep learning of nonlinear partial differential equations (2018)
36. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics informed deep learning (part ii): Data-driven discovery of nonlinear partial differential equations (2017)
37. Ranjan, A., Black, M.J.: Optical flow estimation using a spatial pyramid network (2016)
38. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation (2015)
39. Rudy, S.H., Brunton, S.L., Proctor, J.L., Kutz, J.N.: Data-driven discovery of partial differential equations (2016)
40. Sanchez-Gonzalez, A., Heess, N., Springenberg, J.T., Merel, J., Riedmiller, M., Hadsell, R., Battaglia, P.: Graph networks as learnable physics engines for inference and control (2018)
41. Seo, S., Liu, Y.: Differentiable physics-informed graph networks (2019)
42. Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.k., Woo, W.c.: Convolutional lstm network: A machine learning approach for precipitation nowcasting (2015). https://doi.org/10.48550/ARXIV.1506.04214, https://arxiv.org/abs/1506.04214
43. Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.k., Woo, W.c.: Convolutional lstm network: A machine learning approach for precipitation nowcasting. In: Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1. p. 802–810. NIPS'15, MIT Press, Cambridge, MA, USA (2015)
44. Srivastava, N., Mansimov, E., Salakhutdinov, R.: Unsupervised learning of video representations using lstms (2015). https://doi.org/10.48550/ARXIV.1502.04681, https://arxiv.org/abs/1502.04681

45. van Steenkiste, S., Chang, M., Greff, K., Schmidhuber, J.: Relational neural expectation maximization: Unsupervised discovery of objects and their interactions (2018)
46. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume (2018)
47. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow (2020)
48. Tulyakov, S., Liu, M.Y., Yang, X., Kautz, J.: Mocogan: Decomposing motion and content for video generation (2017)
49. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need (2017)
50. Villegas, R., Yang, J., Hong, S., Lin, X., Lee, H.: Decomposing motion and content for natural video sequence prediction (2018)
51. Vondrick, C., Pirsiavash, H., Torralba, A.: Generating videos with scene dynamics (2016)
52. Wang, Y., Gao, Z., Long, M., Wang, J., Yu, P.S.: Predrnn++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning (2018). https://doi.org/10.48550/ARXIV.1804.06300, https://arxiv.org/abs/1804.06300
53. Wang, Y., Jiang, L., Yang, M.H., Li, L.J., Long, M., Fei-Fei, L.: Eidetic 3d lstm: A model for video prediction and beyond. In: ICLR (2019)
54. Wang, Y., Wu, H., Zhang, J., Gao, Z., Wang, J., Yu, P.S., Long, M.: Predrnn: A recurrent neural network for spatiotemporal predictive learning (2021). https://doi.org/10.48550/ARXIV.2103.09504, https://arxiv.org/abs/2103.09504
55. Wang, Y., Zhang, J., Zhu, H., Long, M., Wang, J., Yu, P.S.: Memory in memory: A predictive neural network for learning higher-order non-stationarity from spatiotemporal dynamics (2018). https://doi.org/10.48550/ARXIV.1811.07490, https://arxiv.org/abs/1811.07490
56. Watters, N., Tacchetti, A., Weber, T., Pascanu, R., Battaglia, P., Zoran, D.: Visual interaction networks (2017)
57. Wu, J., Lu, E., Kohli, P., Freeman, W.T., Tenenbaum, J.B.: Learning to see physics via visual de-animation. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. p. 152–163. NIPS'17, Curran Associates Inc., Red Hook, NY, USA (2017)
58. Wu, Y., Gao, R., Park, J., Chen, Q.: Future video synthesis with object motion prediction (2020)
59. Xu, J., Ni, B., Li, Z., Cheng, S., Yang, X.: Structure preserving video prediction. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1460–1469 (2018). https://doi.org/10.1109/CVPR.2018.00158
60. Xue, T., Wu, J., Bouman, K.L., Freeman, W.T.: Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks (2016)
61. Yin, Y., Le Guen, V., Dona, J., de Bézenac, E., Ayed, I., Thome, N., Gallinari, P.: Augmenting physical models with deep networks for complex dynamics forecasting*. Journal of Statistical Mechanics: Theory and Experiment **2021**(12), 124012 (Dec 2021). https://doi.org/10.1088/1742-5468/ac3ae5, http://dx.doi.org/10.1088/1742-5468/ac3ae5
62. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks (2020)