Predicting future astronomical events using deep learning

Shashwat Singh\textsuperscript{1,a}, Ankul Prajapati\textsuperscript{2,a}, and Kamlesh N. Pathak\textsuperscript{1,a}

\textsuperscript{1}Sardar Vallabhbhai National Institute of Technology, Surat - 395007, India
\textsuperscript{2}Sorbonne Université, Paris - 75005, France
\textsuperscript{a}Bose.X TRIAC\textsuperscript{†}

Abstract

In a quest towards an intelligent decision-making machine, the ability to make plausible predictions is the central pillar of its intelligence. A predicting algorithm’s central idea is to understand the governing physical rules and make plausible and apt predictions based on the same governing laws. Extending the study towards the astrophysical phenomenon puts the model’s ability to test since the model has to understand various parameters that govern the dynamics of the event and understand the spatial and temporal evolution by applying the plausible laws. This work presents a deep learning model to predict plausible future events that maintain spatial and temporal coherence. We have trained over two broad classes, the evolution of $\text{Sa}$, $\text{Sb}$, $\text{S0}$, and $\text{Sd}$ galaxy mergers and evolution of gravitational lenses with a higher redshift of the foreground galaxy having $15M_\odot$. We extended our work towards developing a direct measure of the performance metric for any prediction algorithm. We thereby introduce a novel metric, Correctness Factor (CF), which directly outputs how accurate a prediction is.

1 Introduction

A theoretical framework, if correct, is sufficient enough to predict events in the near future and at the cosmic scale. It even enables to trace back and reason the circumstances that lead to the existing conditions. Thus it tends to develop a deterministic approach towards studying the cosmos. However, fluctuations at the minute level can lead to a domino wherein a completely different condition unfolds. This deviation from the prediction is extremely important and governs the direction and pace of our understanding. Thus astrophysical predictions have always been crucial but are extremely challenging essentially due to the two primary factors: rarity of the event and cosmic timescales at which these events occur. Secondly, even if any event occurs, the probability of a successful observation largely depends on the instrument’s accuracy and precision. A well established theoretical framework can assist in the same in a sense by creating an alert prior to the event.

Astronomical data is usually a time-series sequence of particular spatial parameters that evolves during the given time sampled. A 1-dimensional signal is a time series evolution of certain fixed parameters evolving over discrete time samples. For a 2-dimensional sequence, which is the current focus of our work, is a spatiotemporal sequence wherein the data has space and time components. The quality of data and information corresponding to the data increases significantly on further increasing the dimensions. In terms of Deep Learning (DL), such sequence prediction problems are dealt with by learning from past events and making predictions based on such events. However, such problems are nontrivial due to the

\textsuperscript{*}shashwat98singh@gmail.com
\textsuperscript{†}https://www.bose.org/
high dimensionality of the spatiotemporal sequences, especially when multi-step predictions have to be made. Moreover, building an effective prediction model for astronomical data is even more challenging due to the chaotic nature that governs the astrophysical systems’ dynamics.

In this work of ours, we present a DL model that has been successfully employed in making astronomical prediction over two broad classes: Sa, Sb, S0, and Sd galaxy mergers and evolution of gravitational lenses with a higher redshift of the foreground galaxy. Working along the same lines, we also propose a novel metric, the Correctness Factor (CF), which directly measures how accurate a prediction is, bridging the gap between the mathematical formalism for metric for a spatiotemporal sequence-prediction requiring the least human interference. The paper has been structured such that Sec-2 develops a general overview of the problem and how we attempt to deal with the same. Sec-3 discussing the previous works in the field, Sec-4 is devoted to giving insights to our datasets and preparation methodology. In Sec-5 we present the DL model that has been employed for predicting spatiotemporal sequence, and Sec-6 we give insights to how we concluded our metric and validate it with test and experimentations in Sec-7 and Sec-8. Finally, we conclude our results in Sec-9.
2 Overview

2.1 Problem Statement

In this section, we shall discuss the problem by focusing on a set of spatiotemporal sequences extracted from our dataset. Each data point in the sequence is an arrangement of $M \times N$ grid consisting of $M$ rows and $N$ columns with $C$ color channels. Thus, the observation at any time can be represented by a tensor $X \in R^{M \times N \times C}$, where $R$ denotes the domain of the observed features. If we record the observations periodically, we will get a sequence of tensors $X_1, X_2, \ldots, X_t$ (representation in Fig-2 and Fig-1). The spatiotemporal sequence prediction problem is to predict the most probable length-$K$ sequence in the future given the past $J$ observations, including the current one.

$$\bar{X}_{t+1}, \ldots, \bar{X}_{t+K} = \arg \max_{X_{t+1}, \ldots, X_{t+K}} p(X_{t+1}, \ldots, X_{t+K} | \hat{X}_{t-J+1}, \hat{X}_{t-J+2}, \ldots, \hat{X}_t)$$

2.2 Solution proposed

The underlying idea of predicting future events is drawn from the cognitive neuroscience field [14]: the human mind established complex neural pathways of the physical and causal rules governing the environment primarily through observation and interaction [6]-[1]. The brain is continuously learning intuitive physics, to name a few: trajectories of a falling ball, flow of fluids, collision physics, and refines the already understood world models from the mismatch between its predictions and what truly occurred. Similarly, the deep learning model is trained on historic data wherein it attempts to find patterns through which it makes a plausible prediction having a physical significance. Although seemingly effortless, the idea is more challenging with the astronomical data.

For the model to make robust predictions from the sequence with spatial and temporal components, it needs to be efficient enough to extract meaningful features, e.g., in our case: how much galaxy is sheared during the merger, view at different angles, the evolution of multiple images due to gravitational lenses and similar such features along with maintaining coherence. Focussing on a small localized part of the input $X_i$ for the given time sequence, we could identify myriad visually similar deformations locally due to the temporal coherence, as shown in Fig-2 and Fig-1. Whereas observing the consecutive frames as a whole would give a visually different but semantically coherent sequence. This inconsistency in the visual appearance of 2D stacked data on various scales is mostly attributed, among other factors, to occlusions and shifts in lighting conditions [14]. Predictive models are able to derive representative spatiotemporal correlations describing the dynamics of the spatiotemporal sequence from this source of temporally organized visual cues.

In our case, the future event prediction task can be associated with a supervised learning approach because the target frame itself acts as a label. Thus, learning by prediction is a self-supervised task, filling the gap between supervised and unsupervised learning.

2.3 General problems

Using an example of our data, we showcase how challenging a prediction problem can be for astronomical data. A single 2D data prediction is difficult primarily due to the lack of contextual information and the infinite possibilities of the latent space. A stacked 2D data at discrete time intervals may assert a temporal sequence, narrowing down to a deterministic outcome and reducing prediction space, but the future prediction at every time step is by nature multimodal.

When there exists an equal possibility of any prediction, the deterministic model tends to output prediction close to the mean value of all the possible outcomes. This randomness is visually expressed as

1The resulting sequence can be viewed as 60 frames per second at: [https://github.com/SSingh087/seq-pred](https://github.com/SSingh087/seq-pred)
Table 1: CNN model summary

| Dataset                                      | Input Shape | Total sequence |
|----------------------------------------------|-------------|----------------|
| Galaxy-mergers                              | 60 × 60     | 216            |
| Galaxy-mergers                              | 80 × 80     | 216            |
| Galaxy-mergers                              | 120 × 120   | 216            |
| Galaxy-mergers (2 colormaps)                | 60 × 60     | 2 × 216        |
| Gravitationally lensed galaxy-mergers (2 colormaps) | 60 × 60     | 2 × 216        |
| Gravitationally lensed mono-galaxy (2 colormaps) | 60 × 60     | 2 × 54         |

As deterministic models cannot handle real-world settings characterized by chaotic dynamics, incorporating uncertainty into the model becomes crucial. With an end-to-end model with sufficient data for training, the prediction can be close to the original. One such approach is shown in [16] wherein a considerable amount of radar echo data can be continuously collected so as to reduce the probability space.

3 Related works

Developments in deep learning, especially in Recurrent Neural Networks (RNNs) and Long-Short Term Memory (LSTM), have greatly tackled this problem. Models [1], [5], [9], [11], [7], [17], [19], present some beneficial insights on how to tackle this problem. Since RNNs face vanishing or exploding gradients, using LSTM and GRUs, these problems were mitigated. Shi et al. extended the application of LSTM models to the image [12]. Graves et al. [10] explored multidimensional LSTM and a few stacked recurrent layers to capture abstract spatiotemporal correlations [8]. In [16] the authors presented a ConvLSTM model used for Precipitation Nowcasting, showing that the stacked ConvLSTM layers outperform the Fully Connected (FC)-LSTM. FC-LSTM layer has proven effective for handling temporal correlation but shows redundancy with the spatial component.

The use of Autoencoders has further eased the problem. Autoencoders has two parts the encoder where important information is encoded, capturing the significant features of the datasets, and the decoder that reconstructs the output based on those captured features but the predictions generated are low dimensional and blurry. This low-quality feature reconstruction problem is tackled by state-of-art Adversarial networks introduced by Goodfellow et al. [15]. Drawing inspiration from game theory, the adversarial nets consist of two networks trained simultaneously, both attempting to minimize the loss.

4 Dataset preparation

All our datasets are a spatiotemporal sequence with 10 2D images stacked over in discrete time intervals. The datasets were preprocessed to study the model’s performance towards the different features that were highlighted. We primarily clustered the datasets in clusters of 2, 20, and 60, done for all the datasets discussed below. We also tested by altering the colormap of the dataset to study and verify a new performance metric: Correctness Factor (CF), which is discussed in detail in Sec 6. The details of the number of datasets and the parameters are given in Table 1. This section aims to give a brief regarding the datasets we used to train the model.
4.1 Dataset 1: Galaxy-mergers

We collected pre-simulated galaxy merger datasets available as Galmer dataset\[4\]. The GalMer database is a library of galaxy merger simulations following a statistical approach by performing and analyzing thousands of simulations of interacting pairs, with initial conditions relevant to all epochs of the universe, at different redshifts.

Although the images need to be manually collected, we developed a web scraping method employed for bulk downloading of images with specific parameters. The code is publicly available at Github\[3\]. Our dataset included 10 images from 200\(\)Myr to 700\(\)Myr, restricting our data to orbit-type 1 only. The sole reason for choosing this orbit was due to the Pericentral distance, which is 8Kpc, and motion energy is 0 proving advantageous as giant galaxy usually merges around 700Myr - 750Myr. The 0 motion energy allowed us to capture the cycle with optimum zoom so that the galaxies do not go off the frame the shape evolution (spatial information) is captured. Only Prograde rotation is used, and correspondingly all 4 available inclinations between the galaxies: 0°, 45°, 75°, and 90°. Further, in each simulation sequence, data points are captured at 30° interval starting from 30° and extending to 30° for \(\phi\) and \(\theta\). The sequence is captured at all possible permutations resulting in 9 sets having 10 images.

4.2 Dataset 2: Galaxy-mergers (colormap)

Galmer Dataset with two-color maps: hue, saturation, value (HSV), and twilight has all the spatiotemporal features as of the Galmer dataset.

4.3 Dataset 3: Gravitationally lensed galaxy-mergers

Another dataset that we used for training is the gravitationally lensed image of the merging galaxies. We used the Galmer dataset for simulating the lensed image of the galaxy. The aim was to produce a sequence wherein the foreground galaxy has a higher redshift than the background galaxy. We chose foreground galaxy to have a mass of \(10^{15}M_\odot\) with a redshift, \(z\) of 0.05 and merging galaxy with \(z = 0.02\). The lens model was fixed for all simulations, combining the SIE and SHEAR model with the light model as a combination of two SERSIC ELLIPSE and one NIE model. We have carried our simulation in Python3.0 using library lenstronomy \[2\]. Our code is publicly available on Github\[4\].

4.4 Dataset 4: Gravitationally lensed mono-galaxy

Dwarf galaxies that are free from the gravitational influence of other galaxies are gravitationally lensed using the same methodology discussed for Gravitationally lensed galaxy-mergers; also, the central idea of moving foreground galaxy is captured. Simulation is captured at 30° interval starting from 30° and extending to 90° for \(\phi\) and \(\theta\). The sequence is captured at all possible permutations resulting in 9 sets having 10 images.

5 Proposed Model

In this section, we discuss the DL model that we proposed to overcome the aforementioned problems and predict the spatiotemporal sequence for the datasets discussed in Sec-4. The model architecture is shown in Fig-3 and summarized in Table-2. Although we experimented with various model parameters described in Table-3 and the experimentation results are discussed in Sec-8. This section discusses the

\[http://galmer.obspm.fr/\]
\[https://github.com/SSingh087/seq-pred/tree/main/web-scraping\]
\[https://github.com/SSingh087/seq-pred/blob/main/lensing/code.py\]
Table 2: Model Summary (the model presented here is for Input shape $60 \times 60 \times 3$)

| Layer          | Input Shape | Output Shape | kernel size | padding |
|----------------|-------------|--------------|-------------|---------|
| ConvLSTM       | (60, 60, 3) | (60, 60, 40) | (3, 3)      | same    |
| BatchNormalization | (60, 60, 40) | (60, 60, 40) | (3, 3)      | –       |
| ConvLSTM       | (60, 60, 40) | (60, 60, 40) | (3, 3)      | same    |
| BatchNormalization | (60, 60, 40) | (60, 60, 40) | (3, 3)      | –       |
| ConvLSTM       | (60, 60, 40) | (60, 60, 40) | (3, 3)      | same    |
| BatchNormalization | (60, 60, 40) | (60, 60, 40) | (3, 3)      | –       |
| ConvLSTM       | (60, 60, 40) | (60, 60, 40) | (3, 3)      | same    |
| BatchNormalization | (60, 60, 40) | (60, 60, 40) | (3, 3)      | –       |
| ConvLSTM       | (60, 60, 40) | (60, 60, 40) | (3, 3)      | same    |
| BatchNormalization | (60, 60, 40) | (60, 60, 40) | (3, 3)      | –       |
| Conv3D         | (60, 60, 40) | (60, 60, 3)  | (3, 3, 3)   | same    |

Fundamental model architecture, which on running for 800 epochs, successfully employed frame prediction with minimum loss. From Fig-3, it can be seen that our model follows a sequential approach with tensors passing through various layers sequentially. We avoided the use of Max-pooling or Average-pooling primarily to prevent loss of information since these layers tend to reduce the number of parameters and the computational load. Moreover, the input and output sizes are equivalent. In our model, a Batch Normalization layer follows a ConvLSTM layer, keeping padding as “same” with an argument to return a sequence. The kernel shape and size was fixed for all experiments and activation sigmoid was used for the Conv3D layer with kernel size of $3 \times 3$.

6 Metric for model’s performance

Multiple parameters have to be addressed for a sequence prediction, such as correct temporal prediction, plausible spatial components, brightness, and prediction’s sharpness. The review by Oprea et al. [14] highlighted the absence of evaluation protocols and metrics that evaluate the predictions by simultaneously fulfilling all these aspects. Evaluating a prediction according to the mismatch between its visual appearance and the ground truth is not always reliable because the metrics penalize the predictions that deviate from the ground truth. This results in the predictions being blurry, rather than sharp and bright, as the model attempt to predict values close to the ground truth [3], [13], [20].

Addressing the same issue, we propose a novel performance metric for 2D spatiotemporal sequences, tested for our datasets, and has performed notably. We do expect to have similar performance for other 2D spatiotemporal datasets as well. In our case, a 2D frame is a weighted combination of three color channels - R, G, B. At the end of 800 epochs, model performance was evaluated not solely based on the loss function but a metric that we define as Correctness Factor (CF). CF is a combination of two functions- Root mean square pixel-wise difference (RMSPWD) and Channel-wise Standard deviation (CSD). RMSPWD is the square root difference of the squared pixel values between the ground truth and the predicted sequence for each frame individually for each color channel, and CSD is the difference between the standard deviations of the pixel values of the ground truth and predicted frame corresponding to each channel. In Eq-[6], we define the mathematical approach we used towards the newly defined metric CF.
Figure 3: Model used for predicting spatiotemporal sequence follows a sequential approach with Batch-Normalization layer corresponding to every ConvLSTM layer. The model presented here is for Input shape $60 \times 60 \times 3$.

$$
\begin{align*}
\hat{z}_o^{C_1} &= f(x, y) = \hat{X}_{\text{single color channel}} \\
\hat{z}_p^{C_1} &= g(x, y) = \hat{X}_{\text{single color channel}} \\
\text{RMSPWD} &= \alpha = \sqrt{|z_o^{C_1} - z_p^{C_1}|} \\
\text{CSD} &= \gamma = \sqrt{\frac{\sum_{i=0}^{M\times N} (\hat{z}_o^{C_1} - \frac{\sum_{i=0}^{M\times N} \hat{z}_o^{C_1}}{M\times N})^2}{M\times N}} \\
&\quad - \sqrt{\frac{\sum_{i=0}^{M\times N} (\hat{z}_p^{C_1} - \frac{\sum_{i=0}^{M\times N} \hat{z}_p^{C_1}}{M\times N})^2}{M\times N}}.
\end{align*}
$$

In Eq.6, $\hat{z}_o^{C_1}$ and $\hat{z}_p^{C_1}$ are 3D tensors with symbols as explained in Eq.2.1 wherein $C_1$ represents single color channel. Tensor $z$ can be written as a function of $x$ representing the frame’s width and $y$ the length for an individual frame.

RMSPWD returns the distance between the predicted frame and the ground truth, giving an idea of how far the prediction is from the ground-truth capturing the idea of frame brightness. Using Fig.26 and Fig.27 we demonstrate the RMSPWD in Fig.4 for $M, N = (75, 75)$ evaluated for 10 frames. It can be seen that channels of HSV show extreme scale differences for $M, N = (75, 75)$. R-channel initially has an exact prediction, but CSD increased for higher frames, especially the frames where the foreground galaxy almost crosses over the background galaxy. RMSPWD records such similar trends for all $M, N \in M \times N$ signifying the predicted frame’s sharpness. Also, in Fig.4, we show how the CSD varies for each frame. CSD measures the amount of dispersion of pixel values; the lower the CSD lower the pixel’s deviation from the mean value. For our example, physical significance can be inferred in terms of mass accretion. The higher deviation represents scattered mass throughout the frame, whereas, lower deviation represents

\footnote{The resulting sequence can be viewed as 60 frames per second at: \url{https://github.com/SSingh087/seq-pred}}

\footnote{$M$ and $N$ refers to same terminology as in Eq.2.1}
accreted mass in a localized for fixed angles \((i, \phi, \theta)\). CSD directly measures how blurry the predicted image is by reporting the difference, thereby giving a measure of image sharpness.

We define a function \(\phi(a, \gamma)\) in Eq\((1)\) which outputs a negative log of the combination of the two metrics evaluated per color channel. A minimum of the function will result in the prediction as close to the ground truth. The advantage of the above formalism allows us to establish a metric that combines two fundamental properties of the spatiotemporal prediction: spatial coherence governed by the brightness and temporal components measured by the frame’s sharpness. In Fig\((4)\) we highlight how the the performance is. Choosing \(a\) and \(b\) as unity the range of \(\phi(a, \gamma)\) is \(\kappa \in [-0.5 \log 2, \infty)\). The maximum value of \(\alpha\) and \(\gamma\) can be 1, which shows the maximum difference (deviation from the ground truth). And the minimum value of \(\phi(a, \gamma)\) can be 0, which is the perfect prediction. In Eq\((9)\) we show the condition for the minimum error for a close-to-perfect prediction.

\[
\phi(a, \gamma) = \frac{1}{\sqrt{a \alpha^2 + b \gamma^2}}
\]

All of the above metrics are defined for individual color channels. We define CF in Eq\((2)\) wherein \(\phi^R_i\) represents the \(R\) color channel corresponding to the intersection of \(M^{th}\) row and \(R^{th}\) column resulting in a single pixel. Similarly for \(B\) and \(G\) color channels. The idea is that each color channel represents a latent space of its own, and all the functions that we have defined earlier are evaluated in their individual latent space (Fig\((9)\)). CF combines those latent spaces to output a single-valued metric for an individual frame. The higher the CF more close-to-perfect will be the prediction. The result can be concluded from Fig\((4)\)\((5)\).

\[
CF = \sqrt{\sum_{i=0}^{M \times N} [\phi^R_i(a, \gamma)]^2 + \sum_{i=0}^{M \times N} [\phi^G_i(a, \gamma)]^2 + \sum_{i=0}^{M \times N} [\phi^B_i(a, \gamma)]^2}
\]

---

**Figure 4:** The left column shows the RMSPWD and the right column shows the CSD. This is evaluated in reference to Fig\((26)\) and Fig\((27)\).

---

### 7 Training and testing

We trained our model for 800 epochs using Binary crossentropy\(^7\) as the suitable loss function, with optimizer Adadelta\(^8\).

[7] https://keras.io/api/losses/probabilistic_losses/
[8] https://ruder.io/optimizing-gradient-descent/index.html
8 Experimentation

In this section we discuss our experimentation results on the datasets mentioned in Sec-4 and study its effect on the loss function and CF. The experimentation results are summarized in Table 3.

| Dataset | Varying parameters | Loss function | CF |
|---------|--------------------|---------------|----|
| 1       | Image shape        | Fig-6, Fig-7  | Fig-8 |
|         | Number of Filters  | Fig-10, Fig-11| Fig-12 |
|         | Number of Layers   | Fig-13, Fig-14| Fig-15, Fig-16 |
| 2       | Colormap           | Fig-17, Fig-18| Fig-19 |
| 3       | Colormap           | Fig-20, Fig-21| Fig-22 |
| 4       | Colormap           | Fig-23, Fig-24| Fig-25 |

9 Discussion and Conclusion

In this section, we explain our results and the notable observations for the datasets, as mentioned in Sec-4 and the varying parameters as compiled in Table 3. In dataset-1 (4.1) from Fig-6 we can observe that the loss function decreases rapidly for all the image-size once number of epochs are close to 100 and converges after crossing 600 – 700 epochs. We also observed that the clustering has a significant effect, and a more stable solution was achieved when the number of clusters was 60 or 0. Thus, from our experiments, we concluded that the reduction in the number of clusters did not assisted towards accelerating the accuracy. A plausible explanation can be that the DL learning model predicts various pixel values corresponding to the input frames and attempts to mimic them, reduction in the number of clusters, which is another way of reducing the spread of the pixel values, results in a larger difference between the ground-truth-frame and the predicted frame. We can also observe a general trend with validation loss, which tends to increase and peaks between 30 – 100 and then maintains a similar gradient as that of prediction loss. Although less prominent for image-size 60, it is clearly visible for the other two image-sizes: 80 and 120.

From Fig-7 we can conclude the similar points with additional information of image-size 120 has a more stable solution with the least loss. Fig-8 also echoes the same features. CF highlights individually for each frame and their prediction performance. We observe that for all the cases, image-size 120 has a higher CF score. Although for the case where the number of clusters is 2, a lower CF is observed but
regains a higher score around 4th – 6th frame. A general trend is also observed wherein the CF score decreases rapidly after 4th frame for image-size 60 and 80. A reason that can be accounted for this general trend is after the 4th frame (Fig.1, Fig.2) the galaxy merger cycle reaches a stage wherein the gravitational potential begins shifting so quickly that star orbits are greatly altered and lose trace of their prior orbit. This process is called “violent relaxation” [18]. At this stage, image-size 60 cannot completely capture the stars that deviate from the orbit and resulting in blurred or confused prediction by the model.

We also observed that the activation function does not assist in prediction rather negatively affects the performance. The same keypoint can be highlighted from the Fig.9. Since the image-size, we chose for this experiment set was 60, a decrease in CF score post 4th frame is observed as for the possible reason explained above. Fig.10 highlights the idea that a higher number of filters improve performance. However, we have not tested for the convergence limit where on further increasing the number of filters, the performance stops increasing. Fig.11 bolsters this idea as the rapid fall in loss function for 10 filters is close to 400-500 epochs. One of the advantages of using CF as the metric can be highlighted in Fig.12 which highlights that 10 filters outperform 30 filters for image-size 60. Thus, the loss function is not the sole measure of performance. Fig.14 validates the idea that was stacking ConvLSTM layers improve the performance with Fig.13 validating the idea that 0 and 60 clusters outperform lower clusters. Fig.15 Fig.16 bolsters the same key point.

Further moving on to dataset-2 (Sec.4.2) from Fig.17 frames with no colormap outperforms others. Also from Fig.18 and Fig.19 the same idea can be captured. Twilight colormap does not perform well and has a lower CF, and the same trend is observed with other datasets. The clustering trend is still maintained here. Dataset-3 (Sec.4.3) and dataset-4 (Sec.4.4) highlights the same trend of HSV performing better than the Twilight colormap. For datasets-3, 4, these results can be verified with human observation.

The complete animation for 10 frames is available at Github[7] wherein each result is verified with human observation. CF is a new metric that we propose for dealing with the spatiotemporal sequence. The formulation has been tested on our dataset only, and a further extension and study is proposed and is underway.

[7]https://github.com/SSingh087/seq-pred/blob/main/README.md
Acknowledgements

We would like to thank authors of [14] for compiling the useful models and giving an in-depth insights of the current frame prediction algorithms. We wish to show our gratitude to many people without whom this work would not have been complete. We would also like to acknowledge Project Horizon for maintaining the Galmer database. We also wish to express our gratitude towards all the reviews that were crucial for this work.

References

[1] Rosalind Baker, Matthew Dexter, Tom E Hardwicke, Aimee Goldstone, and Zoe Kourtzi. Learning to predict: Exposure to temporal sequences facilitates prediction of future events. *Vision research*, 99:124–133, 2014.

[2] Simon Birrer and Adam Amara. lenstronomy: Multi-purpose gravitational lens modelling software package. *Physics of the Dark Universe*, 22:189–201, 2018.

[3] Lluis Castrejon, Nicolas Ballas, and Aaron Courville. Improved conditional vrnns for video prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7608–7617, 2019.

[4] IV Chilingarian, Paola Di Matteo, Francoise Combes, A-L Melchior, and Benoit Semelin. The galmer database: galaxy mergers in the virtual observatory. *Astronomy & Astrophysics*, 518:A61, 2010.

[5] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[6] Axel Cleeremans and James L McClelland. Learning the structure of event sequences. *Journal of Experimental Psychology: General*, 120(3):235, 1991.

[7] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[8] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. *arXiv preprint arXiv:1605.07157*, 2016.

[9] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

[10] Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. Multi-dimensional recurrent neural networks. In *International conference on artificial neural networks*, pages 549–558. Springer, 2007.

[11] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[12] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015.

http://www.projet-horizon.fr/rubrique3.html
\[\phi(\alpha, \gamma) = \log \frac{1}{\sqrt{aa^2 + b\gamma^2}} = -\frac{1}{2} \log(aa^2 + b\gamma^2)\]

\[d\phi(\alpha, \gamma) = \frac{\partial \phi}{\partial \alpha} d\alpha + \frac{\partial \phi}{\partial \gamma} d\gamma = 0\]

\[\Rightarrow d\phi(\alpha, \gamma) = \frac{-aa d\alpha - b\gamma d\gamma}{aa^2 + b\gamma^2} = 0\]

\[\Rightarrow aa d\alpha = -b\gamma d\gamma \Rightarrow d\alpha = \frac{f(x, y)df(x, y) - g(x, y)dg(x, y)}{\sqrt{f(x, y)^2 - g(x, y)^2}}\]

(3)

Now, let \(\gamma = \eta_1 - \eta_2 \Rightarrow d\gamma = d\eta_1 - d\eta_2\) Thus on evaluating \(d\eta_1\) and \(d\eta_2\) and using the condition of minima \(aa d\alpha = -b\gamma d\gamma\) we get

\[a[f(x, y)df(x, y) - g(x, y)dg(x, y)] = -b[\gamma(d\eta_1 - d\eta_2)]\]  

(4)
Figure 6: Variation of loss function with number of clusters for each image size

Figure 7: Variation of loss function with image size corresponding to number of clusters
Figure 8: Variation of CF with varying Image-size corresponding to each cluster

Figure 9: Variation of CF with activation function (Relu) for each clusters
Figure 10: Variation of loss function with varying number of filters corresponding to each cluster

Figure 11: Variation of loss function with each cluster corresponding to number of filters

Figure 12: Variation of CF with number of filters

15
Figure 13: Variation of loss function with number of clusters corresponding to number of layers. The upper row represents the variation for Image-size 60 and the lower row for image size 80.

Figure 14: Variation of loss function with number of layers corresponding to each clusters
Figure 15: Variation of CF with number of layers corresponding to each clusters

Figure 16: Variation of CF with number of layers corresponding to each clusters
Figure 17: Variation of loss function with colormap corresponding to each clusters

Figure 18: Variation of loss function with each clusters corresponding to the two colormaps

Figure 19: Variation of CF with colormap corresponding to each clusters
Figure 20: Variation of loss function with colormap corresponding to each clusters

Figure 21: Variation of loss function with each clusters corresponding to the two colormaps

Figure 22: Variation of CF with colormap corresponding to each clusters
Figure 23: Variation of loss function with colormap corresponding to each clusters

Figure 24: Variation of loss function with each clusters corresponding to the two colormaps

Figure 25: Variation of CF with colormap corresponding to each clusters
Figure 26: Predicted frame and Ground-truth frame plotted side-by-side for all 10 framed for HSV colormap

Figure 27: Predicted frame and Ground-truth frame plotted side-by-side for all 10 framed for TWILIGHT colormap

Figure 28: 10 frames of each color channel - R, G, B of a TWILIGHT colormap highlighting the idea of each colormap as a latent space. We evaluate CF from Eq. combining all the latent space.