Decomposing camera and object motion for an improved video sequence prediction

Meenakshi Sarkar* MEENAKSHISAR@IISC.AC.IN
Debasish Ghose DGHOSE@IISC.AC.IN
Aniruddha Bala ANIRUDDHA127@GMAIL.COM
Department of Aerospace Engineering
Indian Institute of Science
Bangalore, India 560012

Abstract

We propose a novel deep learning framework that focuses on decomposing the motion or the flow of the pixels from the background for an improved and longer prediction of video sequences. We propose to generate multi-timestep pixel level prediction using a framework that is trained to learn the temporal and spatial dependencies encoded in the video data separately. The proposed framework, called Velocity Acceleration Network or VANet, is capable of predicting long term video frames for a static scenario, where the camera is stationary, as well as in dynamic partially observable cases, where the camera is mounted on a moving platform (cars or robots). This framework decomposes the flow of the image sequences into velocity and acceleration maps and learns the temporal transformations using a convolutional LSTM network. Our detailed empirical study on three different datasets (BAIR, KTH and KITTI) shows that conditioning recurrent networks like LSTMs with higher order optical flow maps results in improved inference capabilities for videos.

1. Introduction and Related Works

Prediction is an integral part of our day to day planning and decision making process and it often requires us to understand the complex interactions between the dynamics of various objects in the environment. This is why it is often considered as a fundamental component of intelligence (Bubic et al., 2010). Video prediction often decodes much useful information about the surroundings in a format which is rich in information and can be exploited by learning algorithms. However, the nature of the complex interactions between the dynamics of the different objects in a scene, makes long term video prediction a daunting learning problem (Finn et al., 2016), (Finn and Levine, 2017), (Mathieu et al., 2016), (Villegas et al., 2017), (Gao et al., 2019), (Villegas et al., 2019). Based on the state of the art literature, multi-time step video prediction can be broadly divided into two categories: (i) Video prediction in a fully observable static background where the camera remains still during the course of the recording (Finn and Levine, 2017), (Mathieu et al., 2016), (Villegas et al., 2017); and (ii) Video prediction in dynamic background where the camera is mounted on a moving platform (such as car or a mobile robot). The latter case is often referred to as prediction in partially observable scenario in the literature (Gao et al., 2019), (Villegas

* corresponding author

et al., 2019). The notion of partial observability comes from the continuous occlusion of the background from the motion of the camera.

In the context of automation, planning of different manipulation tasks is often associated with video prediction in a fully observable environment where the camera is fixed and stationary. However, in the case of motion planning problems of autonomous cars and mobile robots, we mostly deal with a partially observable environment as the camera keeps moving. Combining video prediction with model based policy gradient algorithms (Kaiser et al., 2020) or planning algorithms in Hafner et al. (2019), improves sample efficiency of reinforcement learning algorithms by reducing the required number of episodic interactions with the environment compared to other model free methods without compromising performance. Moreover, Ebert et al. (2018) and Dasari et al. (2019) have recently shown how visual predictions can aid robot control problems, especially in unstructured environments.

In the last decade, the major focus of understanding spatio-temporal dynamics of video prediction was mostly confined to the case of fully observable environments with static cameras (Srivastava et al., 2015), (Oh et al., 2015), (Vondrick et al., 2016), (Finn et al., 2016), (Mathieu et al., 2016), (Villegas et al., 2017), (Xu et al., 2018), (Wichers et al., 2018). Most of these frameworks exploit some form of optical flow and content decomposition paradigm to generate pixel level predictions. Many times these predictions were coupled with an adversarial training in order to generate realistic images. However, with the availability of high compute power, there is a recent trend in generating high fidelity video predictions with various generative architectures such as Generative Adversarial Networks (GAN) and Variational Autoencoders (VAE) as given in Liang et al. (2017), Denton and Fergus (2018), Babaeizadeh et al. (2018), Lee et al. (2018), Castrejon et al. (2019), Gao et al. (2019) and Villegas et al. (2019).

A few of these recent works, Gao et al. (2019) and Villegas et al. (2019) tried to address the partial observability problem in dynamic scene prediction with the 'hallucination' powers of the generative (GAN and VAE) models. While these frameworks seem to generate realistic predictions for the moving camera problem, their accuracy comes at the cost of high onboard compute capabilities, a luxury that most small or medium scale robots cannot afford. Instead of using stochastic frameworks, we focus on addressing this problem by observing the physics of motion in two different inertial frames: one associated with the moving camera and the other one associated with the dynamic objects in the scene. Our framework is designed with the simple idea of using the relative velocity of the object as it appears in the inertial frame associated with the camera. For a motion planning problem, the realistic approximation of the scene in the background does not play any significant role. However, performance of the motion planner or the policy generator would largely depend upon how accurately we can approximate the relative motion of the objects in the scene with respect to the camera. Objects would appear to move faster or slower than their original velocities in the image frames as the velocity of the camera influences the relative velocities of the objects. This observation led to the idea of decomposing the flow of the pixels into two different components of velocity and acceleration. Previous works (Villegas et al., 2017) on decomposition of video sequences into motion and content, stopped at the point of using the velocity maps or first order pixel difference maps of two consecutive frames. Those frameworks work well for fully observable scenarios where the camera is stationary. However, when the recording agent itself is dynamic, we need to decompose the

motion further into the second order pixel difference maps that we refer to as acceleration maps along with the velocity maps. Deterministic models often suffer from the problem of collective averaging of predicted pixels values resulting in blurry image frames compared to their stochastic counterparts. However, unlike VAEs, deterministic models do not require large computational resources which makes them suitable for small scale robotic applications and this is why we propose to study the comparative performance of the proposed second order deterministic visual prediction model with stochastic frameworks such as SVAP (Lee et al., 2018) and SVG (Denton and Fergus, 2018)
In this paper we have conducted a fairly extensive empirical study on the performance of our generalised physics based deterministic prediction framework, VANet, and compared it to the state of the art generative architectures in the context of a generalised problem of video prediction in a partially observable environment. We also propose a new generalised loss function that helps the deep frameworks learn to reconstruct the velocity and acceleration maps associated with each video frame.

2. Video Prediction with Proposed VANet

For a generalised set up, where the camera is mounted on a dynamic platform moving on a smooth trajectory, the relative velocity vector of any object appearing on the camera frame gets modified continuously. We intend to capture the dynamics of this changing relative velocity vector with a first order pixel difference or velocity map and a second order pixel difference map that we call acceleration map. Thus, we need 3 consecutive image frames $(\mathbf{x}_t, \mathbf{x}_{t-1}, \mathbf{x}_{t-2})$ at timestep $t, t-1$ and $t-2$ to make prediction of the future frame $\mathbf{x}_{t+1}$ at timestep $t+1$, where $\mathbf{x}_t \in \mathbb{R}^{w \times h \times c}$ represents the image frame at time t with dimension $w \times h \times c$. Due to the physics based design of our framework our network is highly interpretable. Our framework can be thought of as the next generation and more improved version of the Motion and Content network (MCNet) proposed by Villegas et al. (2017). While MCNet pioneered the idea of disentangling first order pixel difference map from images sequences with a motion encoder for unsupervised video prediction, we further generalised it to incorporate the complex interactions between the dynamics of the camera and object inertial frames. The entire algorithm which we refer to as the Velocity Acceleration Network (VANet) as shown in figure 1, can be decomposed into the following components:

I. **Velocity Encoder:** The velocity encoder (f^{vel}), parameterised with θ^{vel}, is designed to capture the temporal dependencies embedded in the velocity map of two consecutive image frames, $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$ at time t and $t-1$, respectively. This network takes the velocity map $\mathbf{v}_t = (\mathbf{x}_t - \mathbf{x}_{t-1}) \in \mathbb{R}^{w \times h \times c}$ at time t as input and maps them into two tensors: velocity feature encoding $\mathbf{v}_t^{en} \in \mathbb{R}^{w^{'} \times h^{'} \times c^{'}}$ and the memory cell state $\mathbf{c}_t^{vel} \in \mathbb{R}^{w^{'} \times h^{'} \times c^{'}}$ at time t as:

$$(\mathbf{v}_t^{en}, \mathbf{c}_t^{vel}) = f^{vel}(\mathbf{v}_t, \mathbf{v}_{t-1}^{en}, \mathbf{c}_{t-1}^{vel}; \theta^{vel}) \tag{1}$$

The memory cell state $\mathbf{c}_t^{vel}$ captures the temporal structure embedded in the velocity maps of $\mathbf{v}_{1:t}$. f^{vel} is designed with convolutional LSTM (Shi et al., 2015) networks and in essence, embeds the velocity component of the pixel space into a low dimensional spatio-temporal feature space.

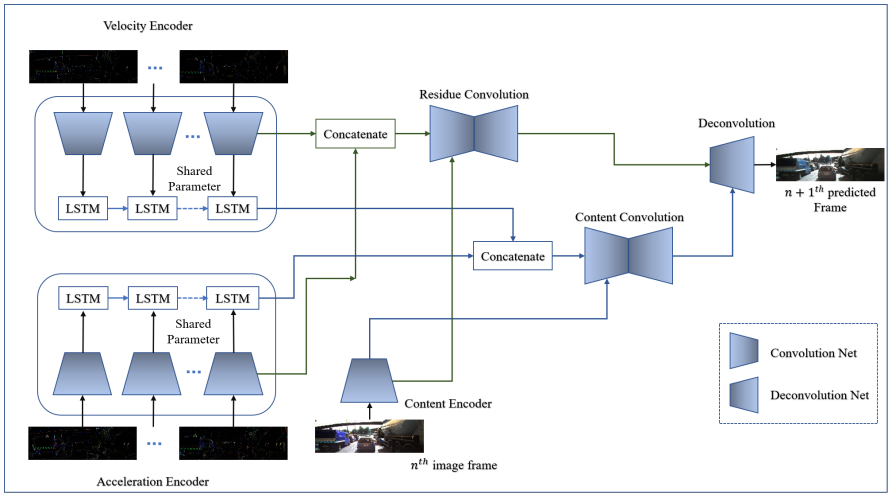


Figure 1: Architecture of VANet while being trained on the KITTI raw dataset. The network learns the temporal dependencies from the velocity and acceleration encoders which takes the first and second order pixel difference maps as inputs, respectively. The content encoder takes the last or n^{th} frame as input to encode the spatial information. Content convolution network combines the spatial encoding with the motion features. Similarly, the residues from the content, velocity and acceleration encoders are fused together in the residue convolution network. Finally, the decoder generates the predicted future frame.

II. **Acceleration Encoder:** The acceleration encoder (f^{acc}) parameterised with θ^{acc} is designed similar to the velocity encoder with only difference of capturing the temporal dependencies embedded in the acceleration map of two consecutive velocity maps, $\mathbf{v}_t$ and $\mathbf{v}_{t-1}$ at time t and $t-1$, respectively. This network takes acceleration map $\mathbf{a}_t = (\mathbf{v}_t - \mathbf{v}_{t-1}) \in \mathbb{R}^{w \times h \times c}$ as input and generates two tensors: acceleration feature encoding $\mathbf{a}_t^{en} \in \mathbb{R}^{w^{'} \times h^{'} \times c^{'}}$ and the memory cell state $\mathbf{c}_t^{acc} \in \mathbb{R}^{w^{'} \times h^{'} \times c^{'}}$ at time t as follows:

$$(\mathbf{a}_t^{en}, \mathbf{c}_t^{acc}) = f^{acc}(\mathbf{a}_t, \mathbf{a}_{t-1}^{en}, \mathbf{c}_{t-1}^{acc}; \theta^{acc}) \tag{2}$$

The memory cell state $\mathbf{c}_t^{acc}$ captures the temporal structure embedded in the acceleration maps of $\mathbf{a}_{1:t}$. This encoder is also designed with convolutional LSTM networks and maps the acceleration component of the pixel space into a low dimensional spatio-temporal latent space.

III. **Content Encoder:** The content encoder (f^{con}), parameterised with θ^{con}, is designed to encapsulate the spatial information embedded in the latest image frame $\mathbf{x}_t$ with a convolutional neural network. The idea here is to map the high dimensional image frames $\mathbf{x}_t \in \mathbb{R}^{w \times h \times c}$ into a low dimensional spatial feature embedding $\mathbf{x}_t^{en} \in \mathbb{R}^{w^{'} \times h^{'} \times c^{'}}$. Mathematically, it can be represented as:

$$\mathbf{x}_t^{en} = f^{con}(\mathbf{x}_t; \theta^{con}) \tag{3}$$

IV. **Content Convolution Network:** This is the part where we start combining the spatial encoding $\mathbf{x}_t^{en}$ coming from the content encoder network with the motion encoding of $\mathbf{v}_t^{en}$ and $\mathbf{a}_t^{en}$. We first combine velocity and acceleration encoding, $[\mathbf{v}_t^{en}, \mathbf{a}_t^{en}] \in \mathbb{R}^{w^{'} \times h^{'} \times 2c^{'}}$, through convolution operations to create the final relative velocity encoding $\mathbf{v_{rel}}_t^{en} \in \mathbb{R}^{w^{'} \times h^{'} \times c^{'}}$ as:

$$\mathbf{v_{rel}}_t^{en} = f^{motion}([\mathbf{v}_t^{en}, \mathbf{a}_t^{en}]; \theta^{motion}) \tag{4}$$

We then combine the relative velocity encoding $\mathbf{v_{rel}}_t^{en}$ with the spatial feature encoding tensor $\mathbf{x}_t^{en}$ with layers of convolution operation and generate the spatio-temporal feature embedding for the next time-step, $\hat{\mathbf{x}}_{t+1}^{en} \in \mathbb{R}^{w^{'} \times h^{'} \times c^{'}}$ given as

$$\hat{\mathbf{x}}_{t+1}^{en} = f^{conv}([\mathbf{x}_t^{en}, \mathbf{v_{rel}}_t^{en}]; \theta^{conv}) \tag{5}$$

Here, f^{motion} and f^{conv} are both designed using CNN having bottleneck architecture (Hinton and Salakhutdinov, 2006), that first projects tensor pairs $[\mathbf{v}_t^{en}, \mathbf{a}_t^{en}]$ and $[\mathbf{x}_t^{en}, \mathbf{v_{rel}}_t^{en}]$ into a low dimensional feature space and then pulls them back to the original feature space of $w^{'} \times h^{'} \times c^{'}$.

V. **Residue Convolution Network:** The idea of temporal transformation of the residues generated from the f^{con} in order to compensate for the loss of information from mapping the high dimensional image frames $\mathbf{x}_t \in \mathbb{R}^{w \times h \times c}$ to a low dimensional feature space $\mathbf{x}_t^{en} \in \mathbb{R}^{w^{'} \times h^{'} \times c^{'}}$ was introduced in Villegas et al. (2017). We carry forward the same idea of multi-scale motion-content residue network but with a modified relative residue velocity encoding $[\check{\mathbf{v}}_{\mathbf{rel}_t}^{en}]^i \in \mathbb{R}^{w^i \times h^i \times c^i}$ at layer i given as:

$$[\check{\mathbf{v}}_{\mathbf{rel}_t}^{en}]^i = f_{res}^{motion}([\check{\mathbf{v}}_t^{en}, \check{\mathbf{a}}_t^{en}]; \theta_{res}^{motion})^i \tag{6}$$

where, $[\check{\mathbf{v}}_t^{en}]^i$ and $[\check{\mathbf{a}}_t^{en}]^i$ are the residue velocity and acceleration encoding from the i^{th} layer of f^{vel} and f^{acc}, respectively. The relative residue velocity encoding $[\check{\mathbf{v}}_{\mathbf{rel}_t}^{en}]^i$ is then combined with the content residue $[\check{\mathbf{x}}_t^{en}]^i$ generated from the i^{th} layer of f^{con} as:

$$[\mathbf{r}_{t+1}^{en}]^i = f_{res}^{conv}([\check{\mathbf{x}}_t^{en}, \check{\mathbf{v}}_{\mathbf{rel}_t}^{en}]; \theta_{res}^{conv}) \tag{7}$$

Like the content convolution network, f_{res}^{motion} and f_{res}^{conv} also uses the CNN bottleneck architecture to combine the tensor pair of $[\check{\mathbf{v}}_t^{en}, \check{\mathbf{a}}_t^{en}]$ and $[\check{\mathbf{x}}_t^{en}, \check{\mathbf{v}}_{\mathbf{rel}_t}^{en}]$.

VI. **Decoder:** Finally, we up-pool the spatio-temporal feature embedding $\hat{\mathbf{x}}_{t+1}^{en}$ and combine it with the residual encoding of $[\mathbf{r}_{t+1}^{en}]^i$ in a layer-wise manner to generate the final prediction of $\tilde{\mathbf{x}}_{t+1}$. The decoder network g^{dec} maps the reduced dimensional $\hat{\mathbf{x}}_{t+1}^{en} \in \mathbb{R}^{w^{'} \times h^{'} \times c^{'}}$ back into the high dimensional pixel level representation of $\tilde{\mathbf{x}}_{t+1} \in \mathbb{R}^{w \times h \times c}$, which is same as the original image frames, given as:

$$\tilde{\mathbf{x}}_{t+1} = g^{dec}([\hat{\mathbf{x}}_{t+1}^{en}, \mathbf{r}_{t+1}^{en}; \theta^{dec}) \tag{8}$$

where, $\mathbf{r}_{t+1}^{en}$ is a list of all residual encoding from f_{res}^{conv} from all its layers. The decoder, g^{dec} uses deconvolutional neural networks (Zeiler et al., 2011) which basically consists of multiple successive operations of deconvolution, rectification and unpooling. The residual embeddings from f_{res}^{conv} is combined via the connection between the Residual Convolution Network and the decoder in a layer-wise manner. The final output layer is passed through a *tanh* non-linearity.

3. Inference and Training

3.1. Inference of multi-time step prediction

In Section 2 we discussed how to make prediction for immediate future frame $\mathbf{x}_{t+1}$ at time-step t with image frames $\mathbf{x}_t$, $\mathbf{x}_{t-1}$ and $\mathbf{x}_{t-2}$ at timesteps $\{t, t-1, \text{and } t-2\}$ and velocity maps $\mathbf{v}_t$ and $\mathbf{v}_{t-1}$. However, for multi-time step prediction, the network observes the velocity and acceleration maps for the last n frames as the difference between image frames $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$ and velocity maps $\mathbf{v}_t$ and $\mathbf{v}_{t-1}$, where $t \in \{2, n\}$, and we assume $\mathbf{x}_1$ is the first observed frame. From this history of past n frames the velocity and acceleration encoders learns the relative pixel dynamics of the scene and then the final frame $\mathbf{x}_n$ is given as input to the Content Encoder. The network then transforms $\mathbf{x}_n$ into $\tilde{\mathbf{x}}_{t+1}$ with the learned dynamics features. For $t \in [n+1, n+T]$, where T is the desired number of prediction steps, VANet starts using its own prediction as input to generate the velocity and acceleration maps.

3.2. Training and Loss function

Since, VANet shares many structural similarities with the architecture of MCNet in Mathieu et al. (2016), we also divided the loss function $\mathcal{L}$ into two major sub-loss functions as:

$$\mathcal{L} = \alpha \mathcal{L}_{image} + \beta \mathcal{L}_{adv} \tag{9}$$

where, $\mathcal{L}_{image}$ and $\mathcal{L}_{adv}$ constitutes the image loss and loss from adversarial training, respectively, and $\alpha, \beta \in \mathbb{R}^+$. We further subdivide $\mathcal{L}_{image}$ into two components: (i) reconstruction loss $\mathcal{L}_{recon}$ and (ii) the total gradient difference loss $\mathcal{L}_{TGDL}$ as follows:

$$\mathcal{L}_{image} = \mathcal{L}_{recon} + \mathcal{L}_{TGDL} \tag{10}$$

The reconstruction loss $\mathcal{L}_{recon}$ is the total L_p norm distance between the ground truth image frame $\mathbf{x}_{n+i}$ and predicted future frames $\tilde{\mathbf{x}}_{t+i}$ for $i \in \{1, .., T\}$ averaged over the entire dataset of $D = \{x^i_{1,\cdots,n,n+1,\cdots,n+T}\}_{i=1}^N$ and is given by:

$$\mathcal{L}_{recon}(\mathbf{x}_{n+1:n+T}, \tilde{\mathbf{x}}_{n+1:n+T}) = \sum_{i=n+1}^{n+T} \|\mathbf{x}_{n+i} - \tilde{\mathbf{x}}_{n+i}\|^p \tag{11}$$

with $p = 1$ or 2. We introduce the total gradient difference loss ($\mathcal{L}_{TGDL}$) which is further divided into gradient difference loss from the predicted image frames ($\mathcal{L}_{GDL}$) and velocity gradient difference loss from the velocity maps generated from the predicted image frames ($\mathcal{L}_{VGDL}$) as:

$$\mathcal{L}_{TGDL} = \mathcal{L}_{GDL} + \mathcal{L}_{VGDL} \tag{12}$$

where,

$$\begin{aligned}\mathcal{L}_{GDL}(\mathbf{x}, \tilde{\mathbf{x}}) = \sum_{t=n+1}^{n+T} \sum_{i,j}^{w,h} &||\mathbf{x}_{t,i,j} - \ \mathbf{x}_{t-1,i,j}| - |\tilde{\mathbf{x}}_{t,i,j} - \tilde{\mathbf{x}}_{t-1,i,j}||^\lambda + \\ &+ ||\mathbf{x}_{t,i,j} - \mathbf{x}_{t,i-1,j}| - |\tilde{\mathbf{x}}_{t,i,j} - \tilde{\mathbf{x}}_{t,i-1,j}||^\lambda + ||\mathbf{x}_{t,i,j} - \mathbf{x}_{t,i,j-1}| - |\tilde{\mathbf{x}}_{t,i,j} - \tilde{\mathbf{x}}_{t,i,j-1}||^\lambda\end{aligned} \tag{13}$$

Here, $\mathcal{L}_{GDL}$ in Eq. (12) is similar to the $\mathcal{L}_{gdl}$ loss in Villegas et al. (2017) in that it gives an average of the gradient difference loss between the predicted frames and ground truth. However, unlike Villegas et al. (2017), we also include the component of temporal difference loss: $||\mathbf{x}_{t,i,j} - \mathbf{x}_{t-1,i,j}| - |\tilde{\mathbf{x}}_{t,i,j} - \tilde{\mathbf{x}}_{t-1,i,j}||^{\lambda}$ in the expression of $\mathcal{L}_{GDL}$ in Eq. (13), so that the velocity encoder can learn the pixel dynamics more efficiently. Here, λ can be chosen to be 1 or 2. Further,

$$\mathcal{L}_{VGDL}(\mathbf{x}, \tilde{\mathbf{x}}) = \sum_{t=n+1}^{n+T} \sum_{i,j}^{w,h} ||\mathbf{v}_{t,i,j} - \ \mathbf{v}_{t-1,i,j}| - |\tilde{\mathbf{v}}_{t,i,j} - \tilde{\mathbf{v}}_{t-1,i,j}||^{\lambda} + \\ + ||\mathbf{v}_{t,i,j} - \mathbf{v}_{t,i-1,j}| - |\tilde{\mathbf{v}}_{t,i,j} - \tilde{\mathbf{v}}_{t,i-1,j}||^{\lambda} + ||\mathbf{v}_{t,i,j} - \mathbf{v}_{t,i,j-1}| - |\tilde{\mathbf{v}}_{t,i,j} - \tilde{\mathbf{v}}_{t,i,j-1}||^{\lambda} \tag{14}$$

The expression for velocity gradient difference loss $\mathcal{L}_{VGDL}$ given in Eq. (14) is similar to that of $\mathcal{L}_{GDL}$ in Eq. (13) with the replacement of $\mathbf{x}$ and $\tilde{\mathbf{x}}$ with the ground truth velocity maps $\mathbf{v}$ and predicted velocity maps $\tilde{\mathbf{v}}$, respectively. The $\mathcal{L}_{VGDL}$ loss is designed so that the acceleration encoder can disentangle and approximate the motion of the pixel dynamics to the second order.

Due to the averaging effect to the reconstruction loss $\mathcal{L}_{image}$ and the blurring effects introduced with the convolution operations, we add an adversarial loss $\mathcal{L}_{adv}$ to the total loss $\mathcal{L}$ in Eq. (9). Similar to Mathieu et al. (2016), $\mathcal{L}_{adv}$ is defined as:

$$\mathcal{L}_{adv} = -\log D([\mathbf{x}_{1:n}, G(\mathbf{x}_{1:n})]) \tag{15}$$

where, $[\mathbf{x}_{1:n}]$ is the concatenation of all input images and $G(\mathbf{x}_{1:n}) = [\tilde{\mathbf{x}}_{n+1:n+T}]$ generates the concatenation of all the predicted future images. Let $[\mathbf{x}_{n+1:n+T}]$ be the concatenation of all ground truth images and $D(.)$ be the output from the discriminator network which is trained with the loss function:

$$\mathcal{L}_{dec} = -\log D([\mathbf{x}_{1:n}, [\mathbf{x}_{n+1:n+T}]) - \log(1 - D([\mathbf{x}_{1:n}, G(\mathbf{x}_{1:n})])) \tag{16}$$

4. Experimental Setup and Methodology

In order to test the proposed framework, we tested it with three different task objectives following similar empirical analysis done by Villegas et al. (2019). We have evaluated the performance of the network with respect to the structural integrity of the predicted frames with respect to the ground truth. We also conducted rigorous studies using five different metrics: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM), VGG Cosine Similarity, Fréchet Video Distance (FVD) (Unterthiner et al., 2018).

1. **Object Interaction:** We evaluated our network on the BAIR towel pick dataset given by Ebert et al. (2018) in order to evaluate its performance in different object interaction tasks. This dataset represents the interaction between different objects and a manipulator. Due to the stochastic nature of the interaction between the objects and the manipulator, our deterministic faces some limitation in long term prediction tasks. However, this type of detailed comparative analysis helps to establish a baseline for the limitation of deterministic video prediction networks in comparison to their stochastic neighbors like SVG (Villegas et al., 2019).

2. **Structured Motion:** One of the well established structured motion prediction datasets is the KTH (Schuldt et al., 2004) human action dataset. Although this dataset is recorded in a fully observable setting, tracking the complex human motion for long duration is a daunting task. Thus, comparing the performance of VANet with SVG and MCNet establishes how our proposed framework performs in comparison to the state of the art generative as well as deterministic methods. The analysis of the comparative performance between VANet and MCNet is of particular interest here as it would help us understand whether the proposed generalised approach toward decomposing motion into velocity and acceleration maps provides any improvement or not.

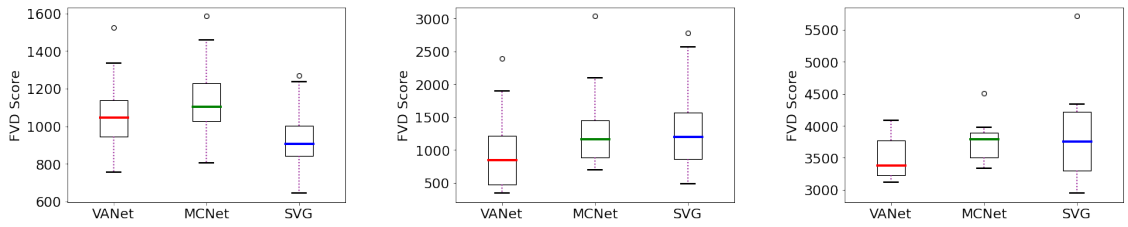
3. **Partial Observability:** Given that the primary objective of the proposed VANet is to generate predictions of object motion in a partially observable scenario, we focus majority of our testing and comparative analysis on this type of tasks. Right now, the only dataset that provides the scope of analysing our network's performance in a partially observable scenario, is KITTI (Geiger et al., 2013) dataset. Here, the video is recorded from a camera fixed on the dashboard of a moving car. Thus, the background of the images keeps getting updated and creates complex interactions between the moving objects on the streets and the relative velocity of the camera.

Ablation Study

We have also conducted a thorough ablation study on our proposed framework by removing various components from it. We have already studied the effects of removing the acceleration encoder from the network when we compare between the performance between VANet and MCNet. However, due to the modular nature of our proposed loss function we can further study the effects of turning off various components of the loss function while back propagating through the network. We trained VANet without the adversarial loss which we refer to as the VANet$_{woGAN}$ to study how adversarial training effects generative capabilities of VANet. For VANet$_{woGAN}$ the loss function in Eq. (9) is simply $\mathcal{L} = \mathcal{L}_{image}$. We also trained VANet$_{NoTD}$ which stands for VANet with no temporal difference loss. In order to train VANet$_{NoTD}$, we modified the expression for $\mathcal{L}_{TGDL}$ to contain only spatial difference terms and no temporal difference terms. For example, we modified $\mathcal{L}_{GDL}$ and $\mathcal{L}_{VGDL}$ from Eq. (13) and Eq. (14) respectively, for VANet$_{NoTD}$ as:

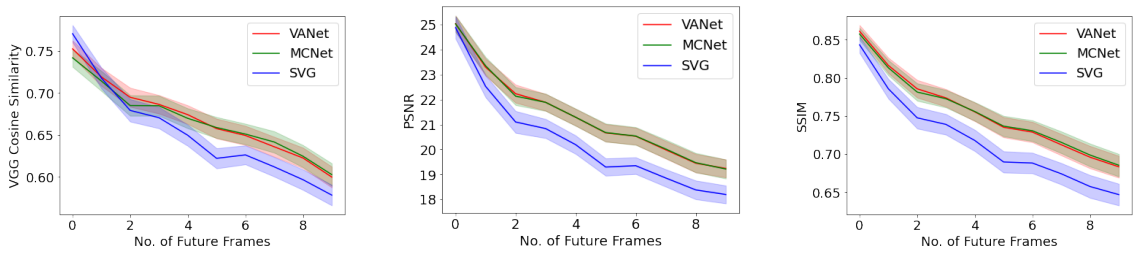
$$\begin{aligned}\mathcal{L}_{GDL}(\mathbf{x}, \tilde{\mathbf{x}}) = \sum_{t=n+1}^{n+T} \sum_{i,j}^{w,h} ||\mathbf{x}_{t,i,j} - \mathbf{x}_{t,i-1,j}| - |\tilde{\mathbf{x}}_{t,i,j} - \tilde{\mathbf{x}}_{t,i-1,j}||^{\lambda} + \\ + ||\mathbf{x}_{t,i,j} - \mathbf{x}_{t,i,j-1}| - |\tilde{\mathbf{x}}_{t,i,j} - \tilde{\mathbf{x}}_{t,i,j-1}||^{\lambda}\end{aligned} \tag{17}$$

$$\begin{aligned}\mathcal{L}_{VGDL}(\mathbf{x}, \tilde{\mathbf{x}}) = \sum_{t=n+1}^{n+T} \sum_{i,j}^{w,h} ||\mathbf{v}_{t,i,j} - \mathbf{v}_{t,i-1,j}| - |\tilde{\mathbf{v}}_{t,i,j} - \tilde{\mathbf{v}}_{t,i-1,j}||^{\lambda} + \\ + ||\mathbf{v}_{t,i,j} - \mathbf{v}_{t,i,j-1}| - |\tilde{\mathbf{v}}_{t,i,j} - \tilde{\mathbf{v}}_{t,i,j-1}||^{\lambda}\end{aligned} \tag{18}$$



(a) (b) (c)

Figure 2: Comparative analysis on the performance of FVD score (lower is better) of VANet, MCNet and SVG on 3 different datasets of BAIR, KTH and KITTI shown respectively



(a) (b) (c)

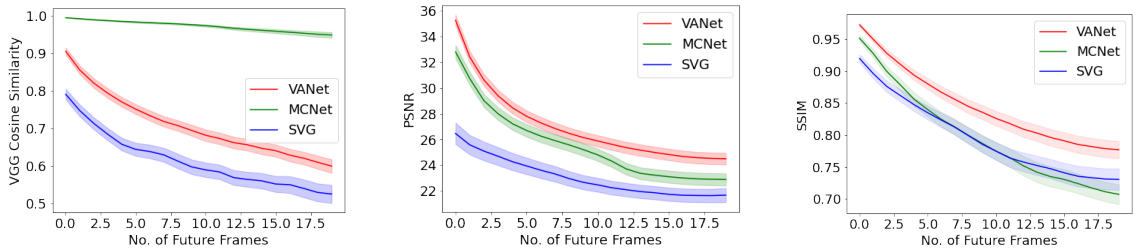
Figure 3: Frame wise quantitative analysis of VANet, MCNet and SVG on BAIR dataset for predicting 10 frames into future conditioned on the past history of 10 frames. We have plotted the mean performance index for VGG Cosine Similarity, PSNR and SSIM (left to right) on the test data-set for each of the networks.

5. Experimental Results

5.1. Object Manipulation with Robotic Arm

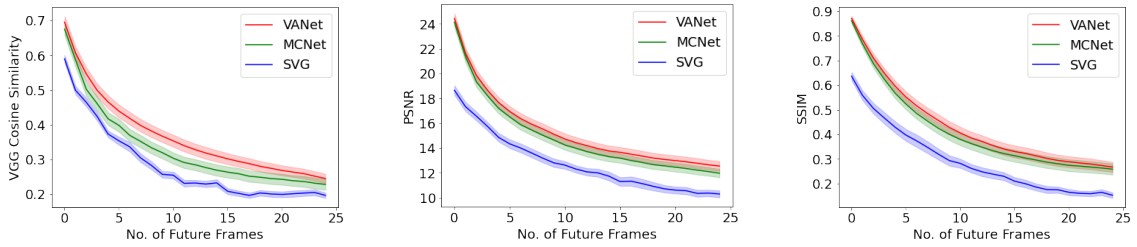
We used the towel pick data-set from Ebert et al. (2018) for capturing the stochastic nature of interaction between a robotic manipulator and various objects. The predicted future frames are conditioned on the action taken by the manipulator. However, in order to capture the true stochastic nature of the predicted future frames, our network is trained to predict future frames conditioned only on the past images frames. This makes our study different from the previous works by Villegas et al. (2019) and Denton and Fergus (2018) where the networks were hard-coded so that the predictions were also conditioned on the action taken by the manipulator. Our predictions are trained to be conditioned on the past 10 frames and predicts 10 future frames at both training and test time. Similar to Villegas et al. (2019) we also resized the original image resolution from 48×64 to 64×64 during training.

The network is trained on 27,744 small clips of a robotic arm picking and placing towel objects in the work-space and tested on 1584 video clips of 20 frames each. We trained with



(a) (b) (c)

Figure 4: Frame wise quantitative analysis of VANet, MCNet and SVG on KTH human action dataset for predicting 20 frames into future based on the past history of 10 frames. We have plotted the mean performance index for VGG Cosine Similarity, PSNR and SSIM (left to right) on the test data-set for each of the networks.



(a) (b) (c)

Figure 5: Frame wise quantitative analysis of VANet, MCNet and SVG on KITTI dataset for predicting 25 frames into future conditioned on the past history of 5 frames. We plotted the mean performance index for VGG Cosine Similarity, PSNR and SSIM (left to right) on the test data-set for each of the networks

Adam optimizer (Kingma and Ba, 2015) with a learning rate of 0.0001, $\beta_1 = 0.5$ and batch size of 8. The other hyper-parameters for the training are chosen as: $\beta = 0.0001$, $\alpha = 1.0$.

Quantitative Evaluation: As discussed in Section 4. we used 4 different evaluation metrics: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM), VGG16 Cosine Similarity, Fréchet Video Distance (FVD), respectively, to provide a detailed comparative study on the performance of VANet with respect to other stochastic as well as deterministic video prediction networks. Since VANet is a direct extension of the previous deterministic framework of MCNet by Villegas et al. (2017), we chose MCNet as the baseline for deterministic video prediction frameworks for our analysis. There are multiple stochastic frameworks already available in the literature such as SVG by Denton and Fergus (2018), SAVP by Lee et al. (2018) and high fidelity stochastic RNN networks by Villegas et al. (2019) for the task of object manipulation. We chose the work on stochastic learned priors by Denton and Fergus (2018) as the baseline for stochastic frameworks, since most of the current state of the art variational

inference frameworks are direct extension of the stochastic video generation work by Denton and Fergus (2018).

Out of the 4 evaluation matrices, FVD measures the spatio-temporal perturbations of the generated videos in its entirety with respect to the ground truth based on the Fréchet Inception Distance (FID). FID is used to evaluate the quality for images from generative frameworks. We presented the box-plots of FVD index from VANet, MCNet and SVG in Fig. 2(a). For frame-wise evaluation we provided the comparative performance plots on VGG16 cosine similarity index in Fig. 3(a), similar to the previous works by Villegas et al. (2019) and Lee et al. (2018). VGG16 cosine similarity index measures the cosine similarity between the flattened high level feature vectors extracted from the VGG network (Simonyan and Zisserman, 2015) provides insights into the differences between the generated and ground truth video frames at the perceptual level. PSNR and SSIM are the most commonly used frame level similarity indexes in the current literature and are plotted in Fig. 3(b) and Fig. 3(c), respectively.

From Fig. 3(a), Fig. 3(b) and Fig. 3(c), it can be seen that at frame level for the towel manipulation task, both the deterministic frameworks VANet and MCNet performs similarly. However, the lower FVD score of VANet compared to MCNet suggests that as a whole the videos generated from VANet are slightly better than the ones generated by MCNet. The similar performace of MCNet and VANet for the object manipulation task was expected for this particular dataset since VANet does not offer any considerable advantage over MCNet when the videos are captured from a stationary platform. It should also be noted from Fig 2(a) that SVG outperforms both MCNet and VANet on the towel picking task since SVG is capable of animating the inherent stochastic nature of the task by learning the associated noise priors.

Qualitative Evaluation: We provide the raw video frames from VANet, MCNet and SVG in Fig. 6 which shows that both VANet and MCNet suffers from blurring effects after the first few predicted frames due to the averaging all future effect ((Villegas et al., 2019) and Denton and Fergus (2018)) of deterministic frameworks. Although in case of VANet the averaging effect is lesser than MCNet.

5.2. Structured Human Motion

Next, we trained VANet, MCNet and SVG on the KTH (Schuldt et al., 2004) human action dataset. We use Adam optimization with learning rate of 0.0001, batch size of 8, $\beta_1 = 0.9$, $\beta = 0.001$ and $\alpha = 1.0$ for training VANet. We train MCNet and SVG on KTH dataset following the hyper-parameters specified in Villegas et al. (2017) and Denton and Fergus (2018), respectively. During training the networks make predictions for the next 10 frames conditioned on 10 frames from the past. During test time the networks generate 20 frames into the future. Each image frame is resized to the resolution of 64×64 during training and inference. The KTH action data-set contains 6 different category of human action: walking, running, jogging, hand-waving, clapping and boxing, respectively, which are divided into training and test sets containing 1528 and 787 small video clips, respectively.

Quantitative Evaluation: The comparative performance-plots on 4 different indexes: FVD, VGG cosine similarity, PSNR and SSIM are shown in Fig. 2(b) and Fig.

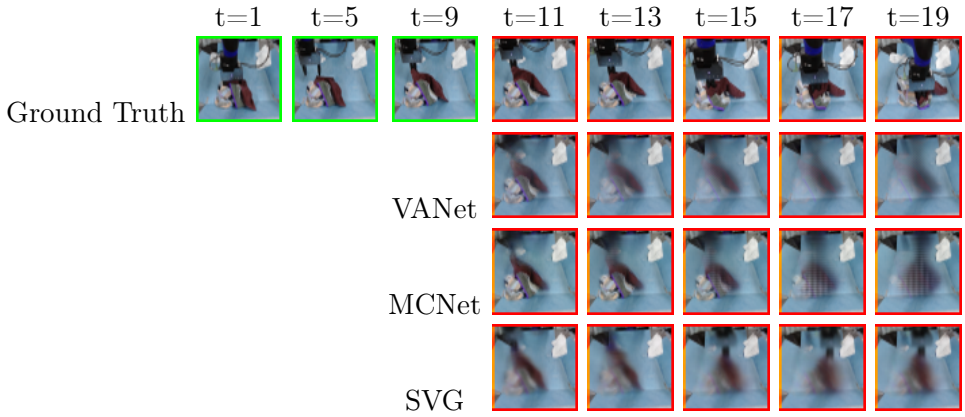


Figure 6: Raw image frames from the predictions on BAIR dataset by VANet, MCNet and SVG network. We provided 10 input frames as history to the network and 10 future frames are predicted.

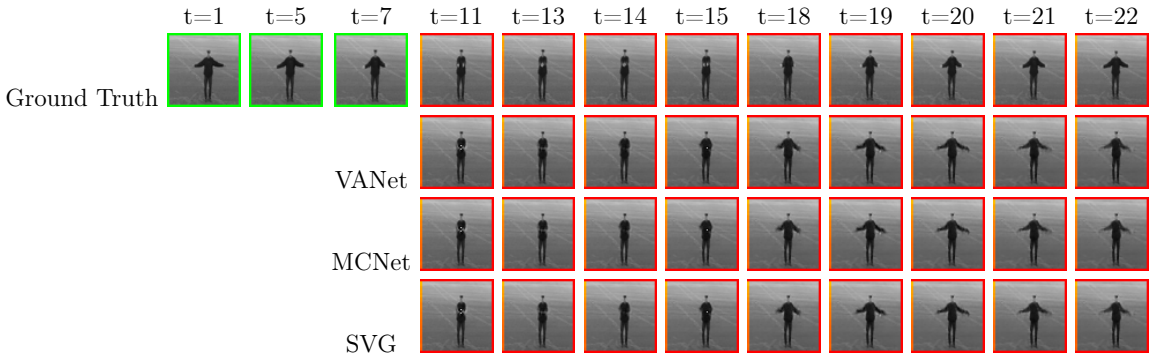


Figure 7: Predictions on KTH Human Action dataset by VANet, MCNet and SVG network. the networks predicts 20 frames into future conditioned on the 10 image frames from the past.

4, respectively. From the plots given in Fig. 4, it can be easily inferred that VANet clearly outperforms MCNet in both frame wise similarity indexes as well as on the overall integrity of generated videos as indicated from the lower FVD score.

Qualitative Evaluation: Examples of raw video frames from the test set generated by VANet, MCNet and SVG are provided in Fig. 7. From Fig. 7, we can see that all the 3 networks produce decent predictions of up to 15 frames into the future from the 10 past input frames. Out of the 3 datasets in our study, KTH human action dataset is the least stochastic one. However, the challenge here is to approximate the complex dynamics between different limbs of the human body involved in the 6 different actions and from Fig. 4 and Fig. 7 we can infer that all the 3 networks perform adequately on this dataset.

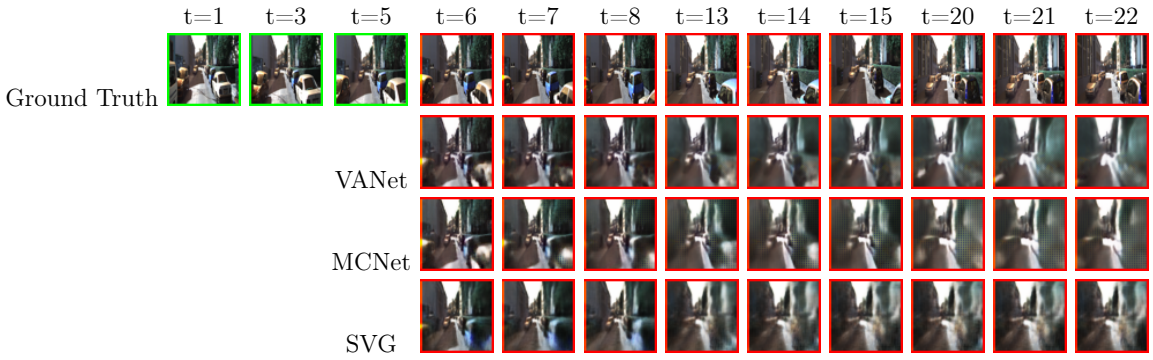


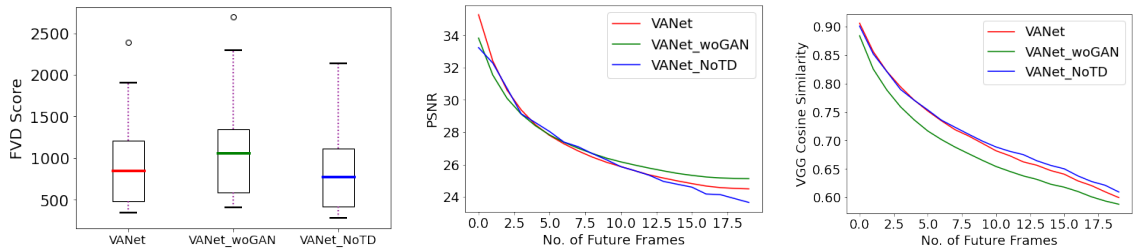
Figure 8: Predictions on partially observable KITTI dataset by VANet, MCNet and SVG. During training the networks were training on 5 input frames and a future prediction of 10 frames. During inference we increase the predicted frames to 25 frames

5.3. Partially Observable Car Driving

Finally, we train our network on the KITTI (Geiger et al., 2013) dataset to test its performance in a partially observable scenario. The dataset contains video clips from the front camera of a car driving down the the streets in Germany. While the original KITTI Raw dataset is much larger, we only selected the videos from the 3 sub-categories of the city, residential neighborhoods, and on the road similar to Villegas et al. (2019). Unlike Villegas et al. (2019) we have taken 50 videos as training set and the remains 11 videos containing clips from all 3 sub-categories are used for testing. These 50 videos are then broken down into smaller video clips of 30 frames each during training and for the test set, the videos are broken down into clips of 40 frames each. We also maintained a gap of 5 frames between each clip during both training and testing time so that each clip is different from the other. We trained the network to predict 10 frames into future from 5 frames from the past. However, during testing we generated 25 frames into future conditioned on the 5 frames from past. We used Adam optimizer with learning rate of 0.0001, batch size of 8, $\beta_1 = 0.9$, $\beta = 0.0001$ and $\alpha = 1.0$. We used these similar hyper-parameters for training MCNet too. For training SVG, we followed the parameter provided by Denton and Fergus (2018) to train the Towel pick dataset.

Quantitative Evaluation: The plots for the quantitative performance analysis on the 4 different indexes of FVD, VGG cosine similarity, PSNR and SSIM are given in Fig. 2(c) and 5, respectively for VANet, MCNet and SVG. Since this is a scenario which encapsulates the complicated interaction between the dynamics of the objects (for instance, other cars on the road) and the camera platform, we can observe the clear difference between the quality of predictions generated by VANet and MCNet in Fig. 5.

Qualitative Evaluation: Examples of raw video frames from the test set generated by VANet, MCNet and SVG are provided in Fig. 8. For Fig. 8, we can see that in case of MCNet, the image quality starts to degrade rapidly after the first 4-5 predicted



(a) (b) (c)

Figure 9: Ablation study of VANet, VANet without adversarial training and VANet without temporal difference loss in $\mathcal{L}_{TGDL}$ on KTH human action dataset for predicting 20 frames into future based on the past history of 10 frames. We have plotted the median FVD scores, mean performance index for VGG Cosine Similarity and PSNR values on the test data-set for each of the networks.

frames. With VANet, this degradation rate is much slower and it can generate decent predictions of up to 15 frames into the future (second row from top in Fig. 8). We also noticed from our simulations that both MCNet and VANet sometimes tend to over estimate the velocity of an upcoming car. However, this over estimation of velocity is rarer and much smaller in case of VANet compared to MCNet.

6. Discussion

The comparative performance analysis between VANet and MCNet clearly shows that VANet outperforms MCNet on both the dynamic evaluation of the entire generated video with lower FVD scores and higher VGG16 cosine similarity, SSIM and PSNR values for better frame-wise structural integrity. The only time MCNet achieves a higher structural similarity score is with the VGG16 cosine similarity index for KTH human action dataset. However, the lower FVD score of VANet and higher SSIM and PSNR values suggests that VANet generates better predictions for the future.

We also point out that from the frame-wise similarity plots given in Fig. 3, Fig. 4 and Fig. 5, it appears that performance of SVG is poorer compared to VANet. We believe this has happened due to insufficient number of training iteration of the network. However, we can see that SVG clearly out-performs the other two networks with a lower FVD score for the towel manipulation task.

7. Ablation Study

As discussed in Section 4 we also conducted an ablation study on the effects of switching off various generative and temporal component of the loss functions given in Eq. (9). A quick review of PSNR values in Fig. 9(b) suggests that all the networks performs similar to each other. However, the relatively lower FVD score of VANet and VANet$_{NoTD}$ in Fig. 9(a) and higher VGG16 similarity index compared to VANet$_{woGAN}$ in Fig. 9(c) suggests that the adversarial training provides considerable improvement in the quality of the predictions.

8. Conclusion

In this paper we have presented a novel physics-based video prediction framework called VANet that exploits first and second order pixel difference flow maps for better approximation of object and camera motion. We have presented a comprehensive study on the performance of VANet compared to two other states of the art stochastic (SVG) and deterministic (MCNet) video prediction framework on three different video datasets. Each of our 3 datasets: BAIR towel pick, KTH human motion and KITTI autonomous car, represents unique challenges for the video predictions task.

The detailed study on the performance of VANet provides an empirical proof that higher order optical flow maps (in this case the first and second order pixel difference maps) can improve the approximating capabilities of the Spatio-Temporal prediction frameworks. This however raises another interesting question on the relation/trade-off between the degree of quantitative improvement in the quality of the generated predictions and the highest order of the optical flow map required by the network.

More details on our project with our code can be found here: https://meenakshisarkar.github.io/Motion-Prediction-and-Planning/vanet/

References

Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H. Campbell, and Sergey Levine. Stochastic variational video prediction. In *Proceedings of the Sixth International Conference on Learning Representations*, ICLR 2018, 2018.

Andreja Bubic, D. Y. Von Cramon, and Ricarda Schubotz. Prediction, cognition and the brain. *Frontiers in Human Neuroscience*, 4:25, 2010.

Lluis Castrejon, Nicolas Ballas, and Aaron Courville. Improved conditional vrnns for video prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, ICCV 2019, October 2019.

Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, Sergey Levine, and Chelsea Finn. Robonet: Large-scale multi-robot learning. In *Proceedings of Conference on Robot Learning CoRL 2019*, volume 100 of *Proceedings of Machine Learning Research*, 2019.

Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. In *Proceedings of the Thirty-fifth International Conference on Machine Learning, ICML 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1174–1183, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex X. Lee, and Sergey Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *CoRR*, abs/1812.00568, 2018.

Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *Proceedings of IEEE International Conference on Robotics and Automation*, ICRA 2017, pages 2786–2793, Singapore, May 2017.

Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *Proceedings of Thirtieth Conference on Neural Information Processing Systems*, NIPS 2016, pages 64–72, Barcelona, Spain, 2016.

Hang Gao, Huazhe Xu, Qi-Zhi Cai, Ruth Wang, Fisher Yu, and Trevor Darrell. Disentangling propagation and generation for video prediction. In *In Proc.. of the IEEE/CVF International Conference on Computer Vision*, ICCV 2019, October 2019.

Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Proceedings of the Thirty-Sixth International Conference on Machine Learning, ICML 2019*, volume 97 of *Proceedings of Machine Learning Research*, pages 2555–2565, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. doi: 10.1126/science.1127647.

Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, Afroz Mohiuddin, Ryan Sepassi, George Tucker, and Henryk Michalewski. Model-based reinforcement learning for atari. In *Proceedings of Eighth International Conference on Learning Representations*, ICLR 2020, Virtual Conference, Formerly Addis Ababa Ethiopia, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

Alex X. Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*, 2018.

Xiaodan Liang, Lisa Lee, Wei Dai, and Eric P. Xing. Dual motion gan for future-flow embedded video prediction. In *Proceedings of IEEE International Conference on Computer Vision*, ICCV 2017, pages 1762–1770, 2017. doi: 10.1109/ICCV.2017.194.

Michaël Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. In *Proceedings of the Fouth International Conference on Learning Representations*, ICLR-2016, San Juan, Puerto Rico, 2016.

Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard L Lewis, and Satinder Singh. Action-conditional video prediction using deep networks in atari games. In *Proceedings of the Twenty-ninth International Conference on Neural Information Processing Systems*, NIPS 2015, pages 2863–2871, Montreal, Canada, 2015.

C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *Proceedings of the 17th International Conference on Pattern Recognition*, volume 3 of *ICPR 2004*, pages 32–36, 2004.

Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai kin Wong, and Wang chun WOO. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Proceedings of the Twenty-ninth International Conference on Neural Information Processing Systems*, NIPS 2015, pages 802–810, Montreal, 2015.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *Proceedings of Thirty-second International Conference on Machine Learning*, ICML 2015, pages 843–852, Lille, France, 2015.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. In *Proceedings of the Fifth International Conference on Learning Representations*, ICLR-2017, Toulon, France, 2017.

Ruben Villegas, Arkanath Pathak, Harini Kannan, Dumitru Erhan, Quoc V Le, and Honglak Lee. High fidelity video prediction with large stochastic recurrent neural networks. In *In Proceedings of the Thirty-second Advances in Neural Information Processing Systems*, NeurIPS 2019, pages 81–91. Curran Associates, Inc., 2019.

Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Anticipating visual representations from unlabeled video. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition CVPR 2016*, pages 98–106, 2016.

Nevan Wichers, R. Villegas, D. Erhan, and H. Lee. Hierarchical long-term video prediction without supervision. In *Proceedings of the Thirty-fifth International Conference on Machine Learning, ICML 2018*, PMLR, 80, pages 6038–6046, Stockholmsmässan, Stockholm Sweden, 2018. PMLR.

Jingwei Xu, Bingbing Ni, and Xiaokang Yang. Video prediction via selective sampling. In *Proceedings of the Thirty-second Conference on Neural Information Processing Systems*, NIPS 2018, pages 1705–1715, Montreal, Canada, 2018.

M. D. Zeiler, G. W. Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *In Proceedings of the IEEE/CVF International Conference on Computer Vision*, ICCV 2011, pages 2018–2025, 2011.